

PmPPC440

PowerPC-Based Processor PMC Module

User's Manual

April 2005



Artesyn Communication Products
8310 Excelsior Dr.
Madison, WI 53717

Web Site:	www.artesyncp.com
Sales:	(800) 356-9602
Technical Support:	(800) 327-1251
PmPPC440 User's Manual—Artesyn Part Number:	10003794-05

PmPPC440

PowerPC-Based Processor PMC Module

User's Manual

April 2005

The information in this manual has been checked and is believed to be accurate and reliable. HOWEVER, NO RESPONSIBILITY IS ASSUMED BY ARTESYN COMMUNICATION PRODUCTS FOR ITS USE OR FOR ANY INACCURACIES. Specifications are subject to change without notice. ARTESYN COMMUNICATION PRODUCTS DOES NOT ASSUME ANY LIABILITY ARISING OUT OF USE OR OTHER APPLICATION OF ANY PRODUCT, CIRCUIT, OR PROGRAM DESCRIBED HEREIN. This document does not convey any license under Artesyn Communication Products patents or the rights of others.

Artesyn and the Artesyn logo are registered trademarks of Artesyn Technologies and are used by Artesyn Communication Products under licence from Artesyn Technologies. All other trademarks are property of their respective owners.

Revision History

Revision Level	Principal Changes	Publication Date	Board Rev.
10003794-00	First release.	February 2003	02
10003794-01	EMI statement, sec. 8.12, figure 3-3	June 2003	02
10003794-02	Correction on page 6-1	September 2003	02
10003794-03	Update Monitor chapter	January 2004	02
10003794-04	Add caution statement in sec. 4	July 2004	02
10003794-05	Add EC declaration, update pages 1-2 and 3-6, Monitor chapter, and product name/subtitle	April 2005	02

Copyright © 2003-2005 Artesyn Communication Products. All rights reserved.

Regulatory Agency Warnings & Notices

The Artesyn PmPPC440 meets the requirements set forth by the Federal Communications Commission (FCC) in Title 47 of the Code of Federal Regulations. The following information is provided as required by this agency.

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

FCC Rules and Regulations – Part 15

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

CAUTION. Making changes or modifications to the PmPPC440 without the explicit consent of Artesyn Communication Products could invalidate the user's authority to operate this equipment.

NOTE. The electromagnetic compatibility (EMC) tests used a PmPPC440 model that includes a front panel assembly from Artesyn Communication Products. For applications where the PmPPC440 is embedded in a system, Artesyn does not provide a front panel, so your system chassis / enclosure must provide the required electromagnetic interference (EMI) shielding.

EC Declaration of Conformity

According to EN 45014:1998

Manufacturer's Name: Artesyn Technologies
Communication Products Division

Manufacturer's Address: 8310 Excelsior Drive
Madison, Wisconsin 53717

**Manufacturer's
Authorized Representative
within the EU/EEA:** Artesyn CP Scandinavia AB
Isafjordsgatan 22,b 5tr
SE-164 40 Kista, Sweden
(Hans Öhman)

Declares that the following product, in accordance with the requirements of 89/336/EEC, EMC directive and 99/5/EC, RTTE directive and their amending directives,

Product: PowerPC-Based Processor PMC Module

Model Name/Number: PmPPC440/10003548-xx

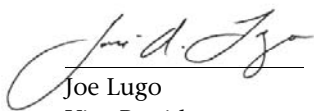
has been designed and manufactured to the following specifications:


EN55022:1998 Information Technology Equipment, Radio disturbance characteristics, Limits and methods of measurement

EN55024:1998 Information Technology Equipment, Immunity characteristics, Limits and methods of measurement

EN300386 V.1.3.1 Electromagnetic compatibility and radio spectrum matters (ERM); Telecommunication network equipment; EMC requirements

As manufacturer we hereby declare that the product named above has been designed to comply with the relevant sections of the above referenced specifications. This product complies with the essential health and safety requirements of the EMC directive and RTTE directive. We have an internal production control system that ensures compliance between the manufactured products and the technical documentation.


Joe Lugo
Vice President
Engineering


Bill Fleury
Compliance Engineer

Issue date: April 19, 2005



Contents

1. Overview

1.1 Components and Features	1-1
1.2 Functional Overview	1-3
1.3 Additional Information	1-4
1.3.1 Product Certification	1-4
1.3.2 Terminology and Notation	1-5
1.3.3 Technical References	1-5

2. Setup

2.1 Electrostatic Discharge	2-1
2.2 PmPPC440 Circuit Board	2-2
2.2.1 Identification Numbers	2-5
2.2.2 Connectors	2-5
2.3 PmPPC440 Setup	2-6
2.3.1 Power Requirements	2-6
2.3.2 Environmental Requirements	2-6
2.3.3 Installing the Module	2-7
2.4 Resets	2-9
2.5 LED Control	2-10
2.6 Monarch Functionality	2-10
2.7 Board Configuration	2-11
2.8 Troubleshooting	2-12
2.8.1 Technical Support	2-12
2.8.2 Service Information	2-13

3. PPC440GP Processor

3.1 On-Chip Features	3-1
3.2 Physical Memory Map	3-2
3.3 On-Card Memory	3-4
3.3.1 User Flash	3-4
3.3.2 SDRAM	3-5
3.3.3 EEPROMs	3-5
3.4 Serial I/O	3-6
3.5 Ethernet	3-6

3.6	I ² C Interface	3-6
3.7	Miscellaneous Status Register	3-7
4.	Real-Time Clock	
4.1	Block Diagram	4-1
4.2	Operation	4-2
4.3	Clock Operation	4-2
5.	PCI/PCI-X Interface	
5.1	PCI Enumeration	5-1
5.2	PCI Reset Control	5-1
5.3	PCI Bus Control Signals	5-2
5.4	Additional PMC Signals	5-4
5.5	PMC Connector Pin Assignments	5-5
6.	Ethernet Interface	
6.1	BCM5221 Registers	6-1
6.2	Ethernet Addresses	6-2
6.3	Front Panel Ethernet Connectors	6-3
7.	Development Mezzanine Card	
7.1	DMC Circuit Board	7-1
7.1.1	Serial Numbers	7-2
7.2	Connectors	7-3
7.2.1	DMC Connector Pin Assignments	7-3
7.2.2	EIA-232 Interface (Unused)	7-6
7.2.3	JTAG/COP Interface	7-7
7.2.4	JTAG Chain Header	7-8
7.2.5	Ethernet Ports (Unused)	7-9
7.3	DMC Jumpers	7-10
7.3.1	Jumper Setting Register	7-10
7.4	Debug/Status LEDs	7-11
7.5	DMC Setup	7-11
7.5.1	Installing the DMC Card	7-12
7.6	Troubleshooting	7-13
7.6.1	Technical Support	7-13
7.7	Service Information	7-13
8.	Monitor	
8.1	Monitor Features	8-1
8.1.1	Start-Up Display	8-1
8.1.2	Auto-repeat	8-2
8.1.3	TFTP booting	8-2
8.1.4	Auto-booting	8-2

8.1.5	Flash Programming	8-2
8.2	Basic Operation	8-3
8.2.1	Power-up/Reset Sequence	8-3
8.2.2	Monitor SDRAM Usage	8-5
8.3	Monitor Recovery	8-6
8.3.1	Corrupted Monitor in Soldered Flash	8-6
8.3.2	Corrupted Bootstrap I ² C Parameters	8-7
8.3.3	Resetting Environment Variables	8-7
8.4	Monitor Command Reference	8-8
8.4.1	Command Syntax	8-8
8.4.2	Typographic Conventions	8-9
8.5	Boot Commands	8-9
8.5.1	bootm (boot memory)	8-9
8.5.2	bootp (boot BootP/TFTP)	8-9
8.5.3	tftpboot (boot TFTP)	8-9
8.5.4	rarpboot (boot RARP/TFTP)	8-10
8.5.5	bootd (boot default)	8-10
8.5.6	bootelf (boot .elf image)	8-10
8.5.7	bootvx (boot VxWorks)	8-10
8.6	File Load Commands	8-11
8.6.1	loads (load S-record)	8-11
8.6.2	loadb (load binary)	8-11
8.7	Memory Commands	8-12
8.7.1	md (memory display)	8-12
8.7.2	mw (memory write)	8-12
8.7.3	mm (memory modify – incrementing address)	8-13
8.7.4	nm (memory modify – constant address)	8-13
8.7.5	cp (copy memory)	8-14
8.7.6	cmp (compare memory)	8-14
8.8	I ² C Device Commands	8-15
8.8.1	imd (I ² C memory display)	8-15
8.8.2	imm (I ² C memory modify – incrementing address)	8-15
8.8.3	inm (I ² C memory modify – constant address)	8-15
8.8.4	imw (I ² C memory write)	8-15
8.8.5	icrc32	8-16
8.8.6	iprobe	8-16
8.8.7	iloop	8-17
8.8.8	isdram	8-17
8.9	Flash Commands	8-18
8.9.1	flinfo (Flash info)	8-18
8.9.2	erase (Flash erase)	8-18
8.9.3	protect (Flash sector protect)	8-19
8.10	Environment Parameter Commands	8-20

8.10.1	printenv (print environment)	8-20
8.10.2	setenv (set environment)	8-20
8.10.3	saveenv (save environment)	8-20
8.10.4	initenv (initialize environment)	8-21
8.11	Test Commands	8-21
8.11.1	um (destructive memory test)	8-21
8.11.2	mtest (memory test)	8-21
8.12	Other Commands	8-21
8.12.1	go	8-21
8.12.2	run	8-22
8.12.3	crc32	8-22
8.12.4	base	8-22
8.12.5	bdinfo	8-22
8.12.6	iminfo	8-22
8.12.7	coninfo	8-23
8.12.8	loop	8-23
8.12.9	reset	8-23
8.12.10	echo	8-23
8.12.11	version	8-23
8.12.12	sleep	8-23
8.12.13	help	8-24
8.13	Environment Variables	8-24
8.14	Troubleshooting	8-25
8.15	Download Formats	8-25
8.15.1	Binary	8-25
8.15.2	Motorola S-Record	8-25

Figures

Figure 1-1.	General System Block Diagram	1-3
Figure 2-1.	PmPPC440 Front Panel	2-2
Figure 2-2.	Component Map, Top (Rev. 02)	2-3
Figure 2-3.	Component Map, Bottom (Rev. 02)	2-4
Figure 2-4.	Module Location on cPCI Baseboard	2-7
Figure 2-5.	Module Location on VME Baseboard	2-7
Figure 2-6.	Installing the Module	2-8
Figure 3-1.	PPC440GP Block Diagram	3-2
Figure 3-2.	PPC440GP Memory Map	3-3
Figure 3-3.	Memory Map for 64-Kilobit EEPROM	3-5
Figure 4-1.	M41T00 Real-Time Clock Block Diagram	4-1
Figure 7-1.	DMC Component Maps, Top and Bottom (Rev. 01)	7-2
Figure 7-2.	DMC PCB-To-PCB Connector, P1	7-6
Figure 7-3.	DMC Mini-B USB Connector, P2	7-6
Figure 7-4.	DMC JTAG/COP Header, P3	7-7
Figure 7-5.	DMC JTAG Chain Header, P4	7-8
Figure 7-6.	DMC RJ45 Connectors, P5-P6	7-9
Figure 7-7.	DMC Jumper Pin Assignments, JP1	7-10
Figure 7-8.	DMC Location on PmPPC440	7-12
Figure 8-1.	Monitor Start-up Display	8-2
Figure 8-2.	Monitor Memory Usage	8-5

Register Maps

Register Map 2-1.	Configuration Command	2-9
Register Map 2-2.	Quick Switch Enable	2-9
Register Map 2-3.	Reset Event	2-10
Register Map 2-4.	LED.	2-10
Register Map 2-5.	Board Configuration	2-11
Register Map 3-1.	Configuration Write-Protect	3-5
Register Map 5-1.	EReady	5-1
Register Map 5-2.	PMC Reset Enable.	5-1
Register Map 7-1.	DMC Jumper	7-10

Tables

Table 1-1.	Regulatory Agency Compliance	1-4
Table 1-2.	Technical References	1-5
Table 2-1.	Circuit Board Dimensions	2-2
Table 3-1.	PPC440GP Address Summary	3-4
Table 3-2.	USB Serial Port Pin Assignments, P2.	3-6
Table 4-1.	RTC Register Map	4-3
Table 5-1.	PMC Connector Pin Assignments, P11–P13.	5-5
Table 5-2.	PMC Connector Pin Assignments, P14	5-6
Table 6-1.	BCM5221 MII Register Map Summary	6-1
Table 6-2.	Ethernet Pin Assignments, P3 & P4	6-3
Table 7-1.	DMC Mechanical Specifications	7-1
Table 7-2.	DMC Connector Pin Assignments, P1	7-3
Table 7-3.	DMC USB Connector Pin Assignments, P2	7-6
Table 7-4.	DMC JTAG/COP Connector Pin Assignments, P3.	7-7
Table 7-5.	DMC JTAG Chain Header Pin Assignments, P4	7-8
Table 7-6.	DMC Ethernet Port Pin Assignments, P5–P6	7-9
Table 8-1.	Standard Environment Variables	8-24
Table 8-2.	Optional Environment Variable ^b	8-25

Overview

The Artesyn PmPPC440 module is a Processor PCI Mezzanine Card (PPMC) based on the IBM PowerPC® 440GP Embedded Processor. It functions as a complete, low-power, processor subsystem that can operate in both PPMC Monarch and non-Monarch modes. The PmPPC440 supports various memory configurations, user flash memory, up to two serial ports, two 10/100BaseTX Ethernet ports, and an optional Development Mezzanine Card (DMC). Some typical applications for the PmPPC440 include protocol processing, packet processing, data filtering, and I/O management.

1.1 Components and Features

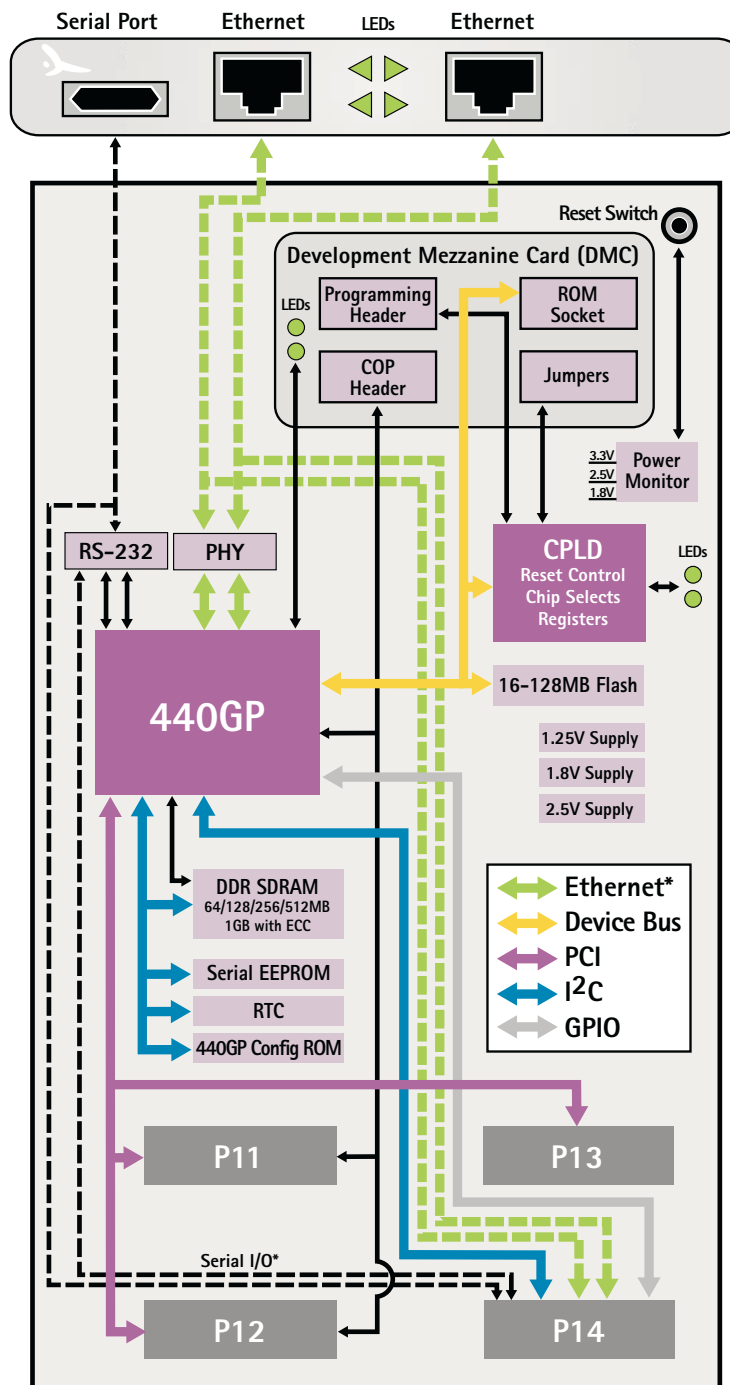
The following is a brief summary of the PmPPC440 hardware components and features:

- | | |
|--------------|--|
| CPU | The central processing unit (CPU) for the PmPPC440 features a PowerPC 440 core, operating at 400MHz. The CPU has 32-kilobyte instruction and data caches. It supports a Double Data Rate (DDR) Synchronous Dynamic RAM (SDRAM) interface, an Inter-Integrated Circuit (I ² C) interface, a PCI-X interface, Ethernet and serial ports, and many other features. |
| SDRAM | The PmPPC440 allows for a 72-bit Small-Outline Dual In-Line Memory Module (SO-DIMM) of up to one gigabyte to support the CPU. This SDRAM operates at 266MHz and has Error-Correcting Code (ECC). The SO-DIMM stores configuration information in its on-board EEPROM, accessible via the I ² C interface. |
| Flash | The PmPPC440 supports up to 128 megabytes of user flash memory. This memory conforms to the Intel StrataFlash™ architecture and is organized as four 16-bit banks on the CPU's device bus. |

I²C Interfaces	The PmPPC440 has an Inter-Integrated Circuit (I ² C) bus controlled by the PPC440GP CPU. One I ² C interface provides software access to the SO-DIMM, the real-time clock (RTC), a 64-kilobit serial EEPROM for non-volatile memory (NVRAM), and a two-kilobit serial configuration EEPROM for the CPU. A second I ² C interface allows for external access via the P14 PMC connector.
Serial I/O	The PPC440GP CPU provides two asynchronous serial ports, which support EIA-232 signal levels. Dual serial ports can be routed to the P14 PMC connector, or a single serial port can be routed to the front panel mini-B Universal Serial Bus (USB) connector.
Ethernet	The PPC440GP CPU provides Media Access Control (MAC) functionality for two 10/100BaseTX Ethernet ports, which can automatically detect the interface speed. Optionally, two ports can be routed to the front panel RJ45 connectors (isolated) or the P14 PMC connector (non-isolated).
CPLD	The PmPPC440 uses a complex programmable logic device (CPLD) to provide many logic functions, including LED control, configuration registers, reset and chip select control, and the DMC interface.
PCI/PCI-X Bus	The PmPPC440 supports a Peripheral Component Interconnect (PCI) / PCI-X bus that can operate at 33/66/133 MHz with a 32- or 64-bit data path. The PmPPC440 can function as a local PCI host (Monarch) or as a peripheral (non-Monarch).
DMC	The Development Mezzanine Card (DMC) is a custom, optional, plug-on card that mounts on the back of the PmPPC440. This card facilitates hardware and software development.

1.2 Functional Overview

The following block diagram provides a functional overview for the PmPPC440:



* Ethernet and Serial I/O configuration options:
 - I/O via front panel
 - I/O via P14 connector

Figure 1-1. General System Block Diagram

1.3 Additional Information

This section lists the PmPPC440 hardware's regulatory certifications and briefly discusses the terminology and notation conventions used in this manual. It also lists general technical references.

1.3.1 Product Certification

The PmPPC440 hardware has been tested and certified to comply with various safety, immunity, and emissions requirements as specified by the Federal Communications Commission (FCC), Underwriters Laboratories (UL), and others. The following table summarizes this compliance:

Table 1-1. Regulatory Agency Compliance

Type	Specification
Safety	IEC950/EN60950 — Safety of Information Technology Equipment (Western Europe) UL60950-1, CSA C22.2 No. 60950-1, 1st Edition — Safety of Information Technology Equipment, including Electrical Business Equipment (BI-National) Global IEC — CB Scheme Report IEC 950, all country deviation
Environmental	NEBS: Telcordia GR-63 — Section 4.1.1 Transportation and Storage Environmental Criteria; Section 4.3 Equipment Handling Criteria; Section 4.4.3 Office Vibration Environment and Criteria; Section 4.4.4 Transportation Vibration Criteria Section 4.5.2 Airborne Contaminants in an Environmentally-Controlled Space Section R4-31 Needle Flame Test
EMC	FCC Part 15, Class B — Title 47, Code of Federal Regulations, Radio Frequency Devices ICES 003, Class B — Radiated and Conducted Emissions, Canada NEBS: Telcordia GR-1089 level 3 — Emissions and Immunity (circuit pack level testing only) ETSI EN300386-2: 1997 — Electromagnetic Compatibility and Radio Spectrum Matters (ERM), Telecommunication Network Equipment, Electromagnetic Compatibility (EMC) Requirements, Part 2: Product Family Standard

Artesyn maintains test reports that provide specific information regarding the methods and equipment used in compliance testing. Unshielded external I/O cables, loose screws, or a poorly grounded chassis may adversely affect the PmPPC440 hardware's ability to comply with any of the stated specifications.

The UL web site at <http://www.ul.com> has a list of Artesyn's UL certifications. To find the list, search in the online certifications directory using Artesyn's UL file number, E190079. There is a list for products distributed in the United States, as

well as a list for products shipped to Canada. To find the PmPPC440, search in the list for 10003548-xx, where xx changes with each revision of the printed circuit board.

1.3.2 Terminology and Notation

Active low signals	An active low signal is indicated with an asterisk * after the signal name.
Byte, word, long word, double long word	Throughout this manual <i>byte</i> refers to 8 bits, <i>word</i> refers to 16 bits, and <i>long word</i> refers to 32 bits, <i>double long word</i> refers to 64 bits.
PLD	This manual uses the acronym, <i>PLD</i> , as a generic term for programmable logic device (also known as FPGA, CPLD, EPLD, etc.).
Radix 2 and 16	Hexadecimal numbers end with a subscript 16. Binary numbers are shown with a subscript 2.

1.3.3 Technical References

Further information on basic operation and programming of the PmPPC440 components can be found in the following documents:

Table 1-2. Technical References

Device or Interface	Type	Document ¹
CPU	PPC440GP	<i>PowerPC 440GP Embedded Processor Data Sheet (Preliminary)</i> (IBM Corporation, SA14-2561-06, March 20, 2002)
		<i>PPC440GP Embedded Processor User's Manual (Preliminary)</i> (IBM Corporation, SA14-2519-11, March 2002)
Ethernet	BCM5221	<i>Book E: Enhanced PowerPC™ Architecture</i> (IBM Corporation, Version 0.90, March 23, 2002) http://www.ibm.com
		<i>BCM5221 10/100Base-TX/FX Mini-φ™ Transceiver Preliminary Data Sheet</i> (Broadcom Corporation, 5221-DS06-R, May 29, 2001) http://www.broadcom.com
		<i>IEEE Standard for Information Technology: IEEE Std 802.3, 2000 Edition</i> (IEEE: New York, NY) http://www.ieee.org

Table 1-2. Technical References — *Continued*

Device or Interface	Type	Document ¹
PCI/PCI-X		<i>PCI Local Bus Specification</i> (PCI Special Interest Group, Revision 2.2, 1998) <i>PCI-X Protocol Addendum to the PCI Local Bus Specification Revision 2.0</i> (PCI Special Interest Group, July 29, 2002) http://www.pcisig.com
PMC		<i>IEEE Standard for a Common Mezzanine Card (CMC) Family: IEEE Std 1386-2001</i> (IEEE: New York, NY) <i>IEEE Standard for Physical and Environmental Layers for PCI Mezzanine Cards: IEEE Std 1386.1-2001</i> (IEEE: New York, NY) http://www.ieee.org
PPMC		<i>Processor PMC Standard for Processor PCI Mezzanine Cards: VITA 32 – 199x, Draft 0.5, May 9, 2002</i> (VITA: Scottsdale, AZ) http://www.vita.com
Serial Interface	EIA-232-F	<i>TIA/EIA-232-F: Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment Employing Serial Binary Data Interchange</i> (Electronic Industries Association, October 1997) http://www.eia.com/

1. Frequently, the most current information regarding addenda/errata for specific documents may be found on the corresponding web site.

If you have questions, please call an Artesyn Product Support Services representative at 1-800-327-1251, visit the web site at <http://www.artesyncp.com>, or send e-mail to support@artesyncp.com.

2

Setup

This chapter describes the physical layout of the boards, the setup process, and how to check for proper operation once the boards have been installed. This chapter also includes troubleshooting, service, and warranty information.

2.1 Electrostatic Discharge

Before you begin the setup process, please remember that electrostatic discharge (ESD) can easily damage the components on the PmPPC440 hardware. Electronic devices, especially those with programmable parts, are susceptible to ESD, which can result in operational failure. Unless you ground yourself properly, static charges can accumulate in your body and cause ESD damage when you touch the board.

CAUTION. Use proper static protection and handle PmPPC440 boards only when absolutely necessary. Always wear a wriststrap to ground your body before touching a board. Keep your body grounded while handling the board. Hold the board by its edges—do not touch any components or circuits. When the board is not in an enclosure, store it in a static-shielding bag.

To ground yourself, wear a grounding wriststrap. Simply placing the board on top of a static-shielding bag does not provide any protection—place it on a grounded dissipative mat. Do not place the board on metal or other conductive surfaces.

2.2 PmPPC440 Circuit Board

The PmPPC440 circuit board is a PCI Mezzanine Card (PMC) assembly. It uses a 12-layer printed circuit board with the following dimensions:

Table 2-1. Circuit Board Dimensions

Width	Length	Height
2.9 in.	5.9 in.	< 0.39 in.
74 mm	149 mm	< 10 mm

The figure below shows the PmPPC440 front panel. The figures on the following pages show the component locations for the PmPPC440 circuit board.

NOTE. The electromagnetic compatibility (EMC) tests used a PmPPC440 model that includes a front panel assembly from Artesyn Communication Products. For applications where the PmPPC440 is embedded in a system, Artesyn does not provide a front panel, so your system chassis/enclosure must provide the required electromagnetic interference (EMI) shielding.

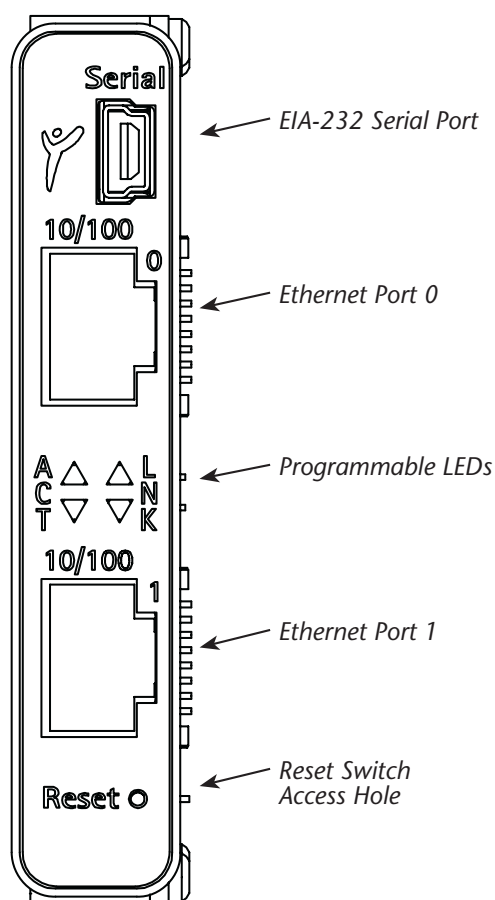
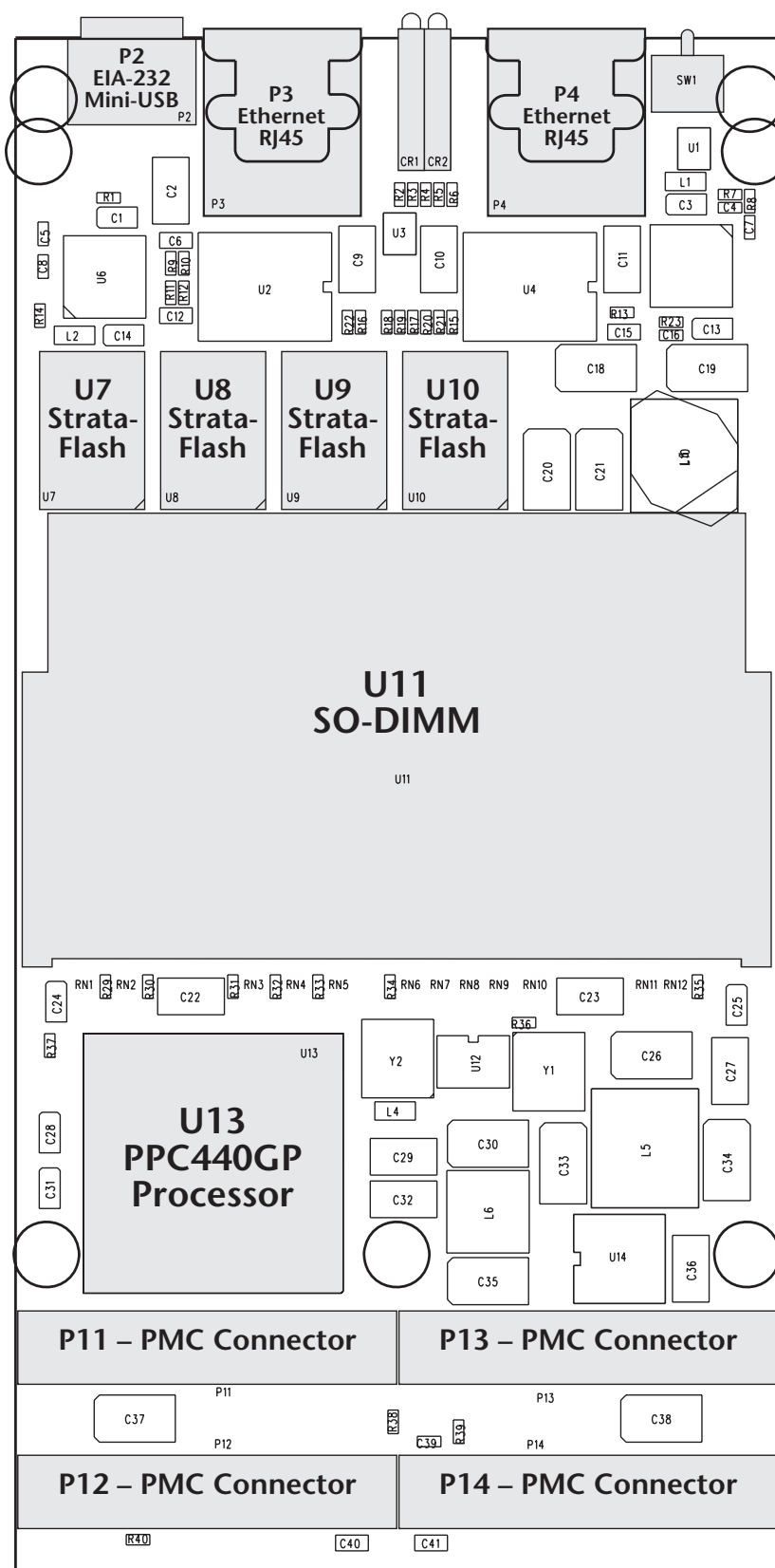


Figure 2-1. PmPPC440 Front Panel



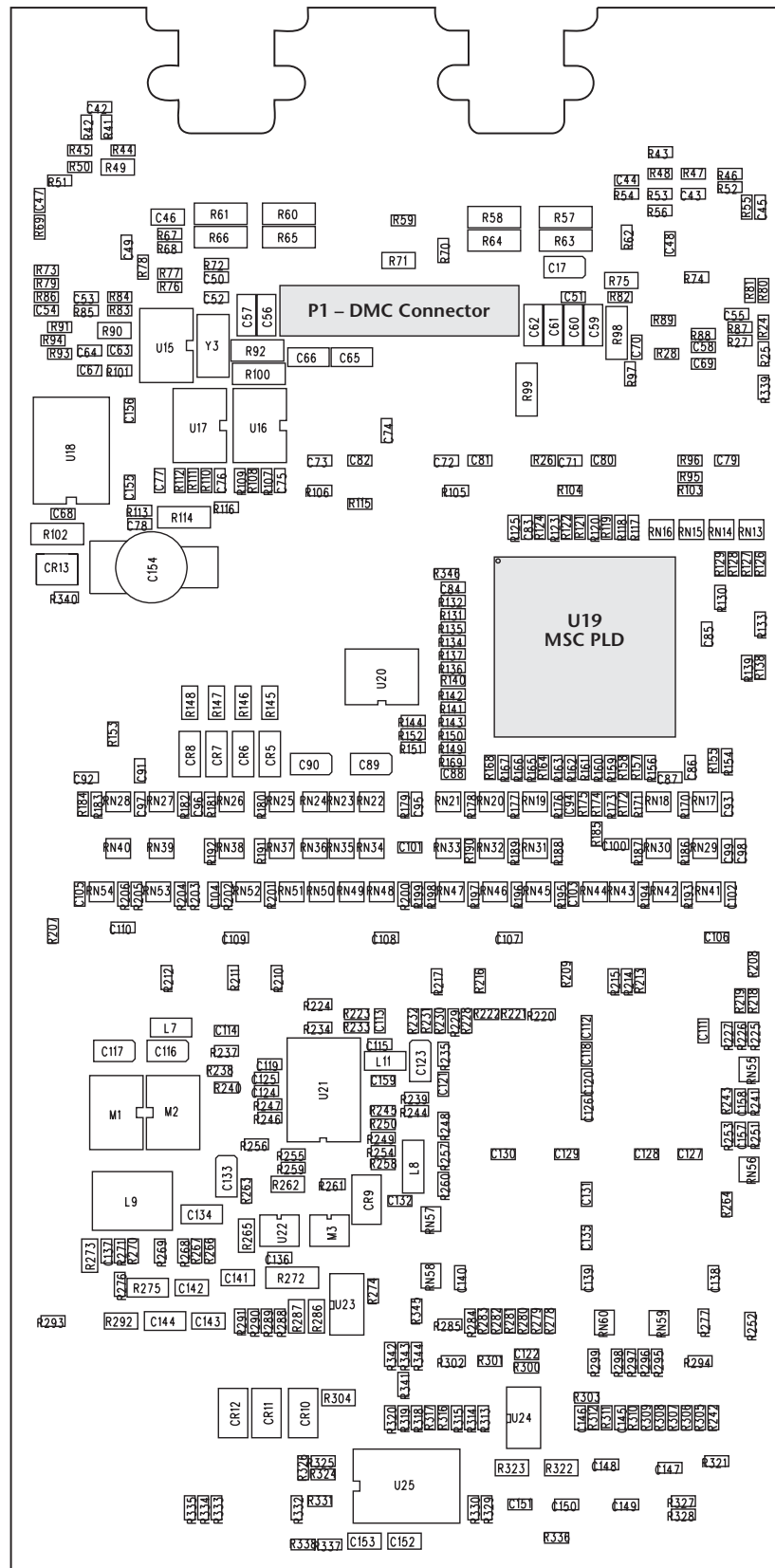


Figure 2-3. Component Map, Bottom (Rev. 02)

2.2.1 Identification Numbers

Before you install the PmPPC440 circuit board in a system, you should record the following information:

- ☐ The board serial number:_____ .

The board serial number appears on a bar code sticker located on the back of the board.

- ☐ The monitor version: _____ .

The version number of the monitor is on the monitor start-up display.

- ☐ The operating system version and part number:_____ .

This information is labeled on the master media supplied by Artesyn or another vendor.

- ☐ The board assembly/revision number: _____ .

A sticker on the board contains the board assembly part number and configuration description. Be sure to include all the information that appears on the sticker.

- ☐ Any custom or user ROM installed, including version and serial number:
_____ .

It is useful to have these numbers available when you contact the Product Support Services department at Artesyn Communication Products.

2.2.2 Connectors

The PmPPC440 circuit board has various connectors, summarized as follows:

- | | |
|----------------|--|
| P1 | Connector P1 is an 80-pin connector on the back side of the circuit board that accepts the optional Development Mezzanine Card (DMC). See Table 7-2 for pinouts. |
| P11–P14 | P11, P12, P13, and P14 are 64-pin PMC connectors that support the PMC slot. See Table 5-1 and Table 5-2 for pinouts. |

2.3 PmPPC440 Setup

You need the following items to set up and check the operation of the Artesyn PmPPC440.

- ☐ Artesyn PmPPC440 board
- ☐ Compatible baseboard host
- ☐ Card cage and power supply
- ☐ Serial interface cable (EIA-232)
- ☐ Terminal

When you unpack the board, save the antistatic bag and box for future shipping or storage.

2.3.1 Power Requirements

The Artesyn PmPPC440 circuit board requires less than seven watts of power. The exact power requirements for the PmPPC440 circuit board depend upon which specific peripheral hardware is installed in the system.

NOTE. The power values given in this manual are approximate—not measured—values. If you have specific questions regarding the board's power requirements, please contact Artesyn Product Support Services at 1-800-327-1251.

2.3.2 Environmental Requirements

The Artesyn PmPPC440 circuit board is specified to operate in an ambient air temperature range of 0° to +55° Centigrade. This range meets and exceeds the NEBS Telcordia GR-63 specification. The entire chassis should be cooled with forced air. The exact air flow requirement depends upon the chassis configuration and the ambient air temperature. The PmPPC440 board's relative humidity and storage temperature ranges fully comply with NEBS Telcordia GR-63 specification.

2.3.3 Installing the Module

PMC-compatible baseboards have one or two sets of four connectors (J11–J14 and J21–J24), as defined by the PMC specification. Fig. 2-4 shows the location of these connectors on a typical cPCI baseboard.

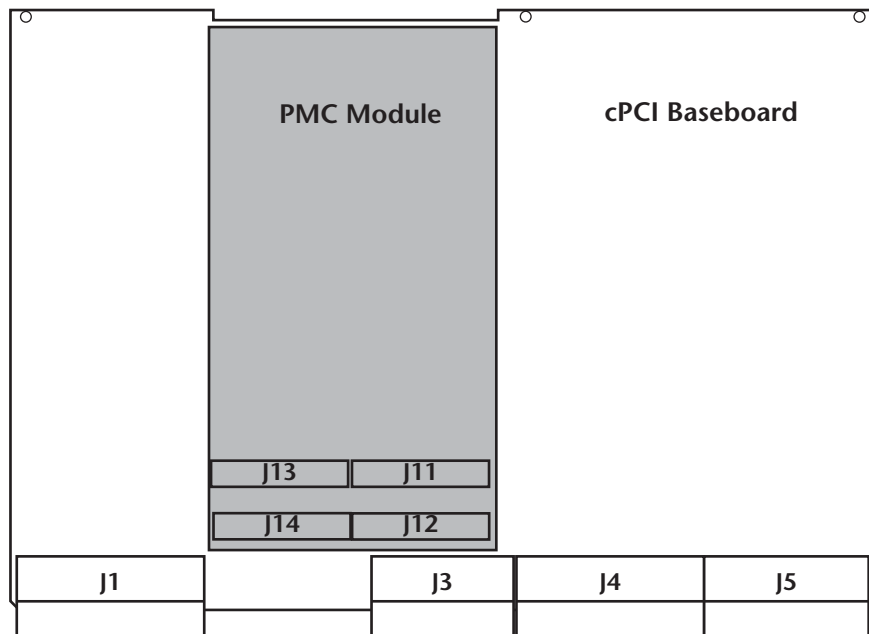


Figure 2-4. Module Location on cPCI Baseboard

Fig. 2-5 shows the location of these connectors on a VME baseboard.

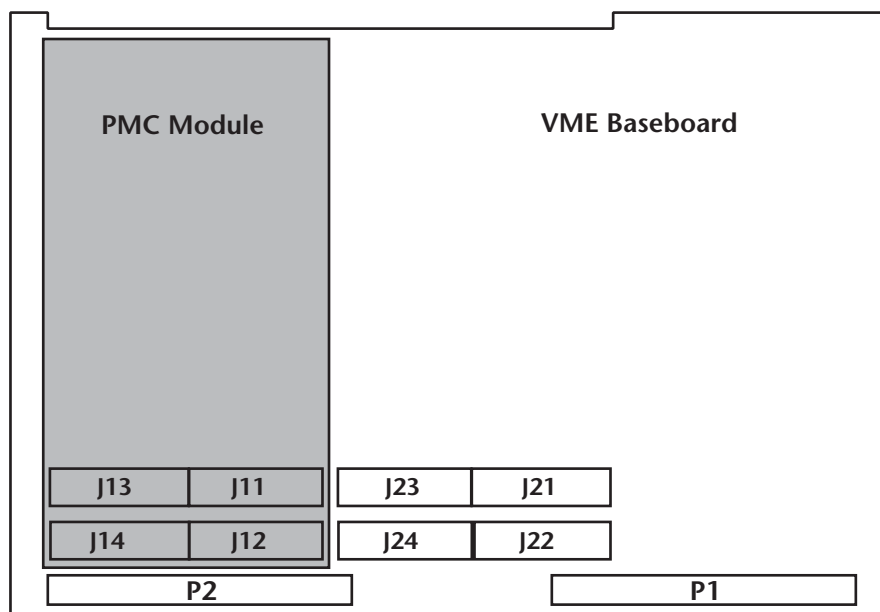


Figure 2-5. Module Location on VME Baseboard

The following procedure describes how to attach the PmPPC440 module to an Artesyn VME baseboard (see Fig. 2-6). For installation on a cPCI baseboard, the procedure would be similar.

1. Remove the screws from the standoffs on the PMC module.
2. Hold the module at an angle and gently slide the faceplate into the opening on the baseboard.
3. Align the P11, P12, P13, and P14 connectors and gently press the module into place until firmly mated.

CAUTION. To avoid damaging the module and/or baseboard, do not force the module onto the baseboard.

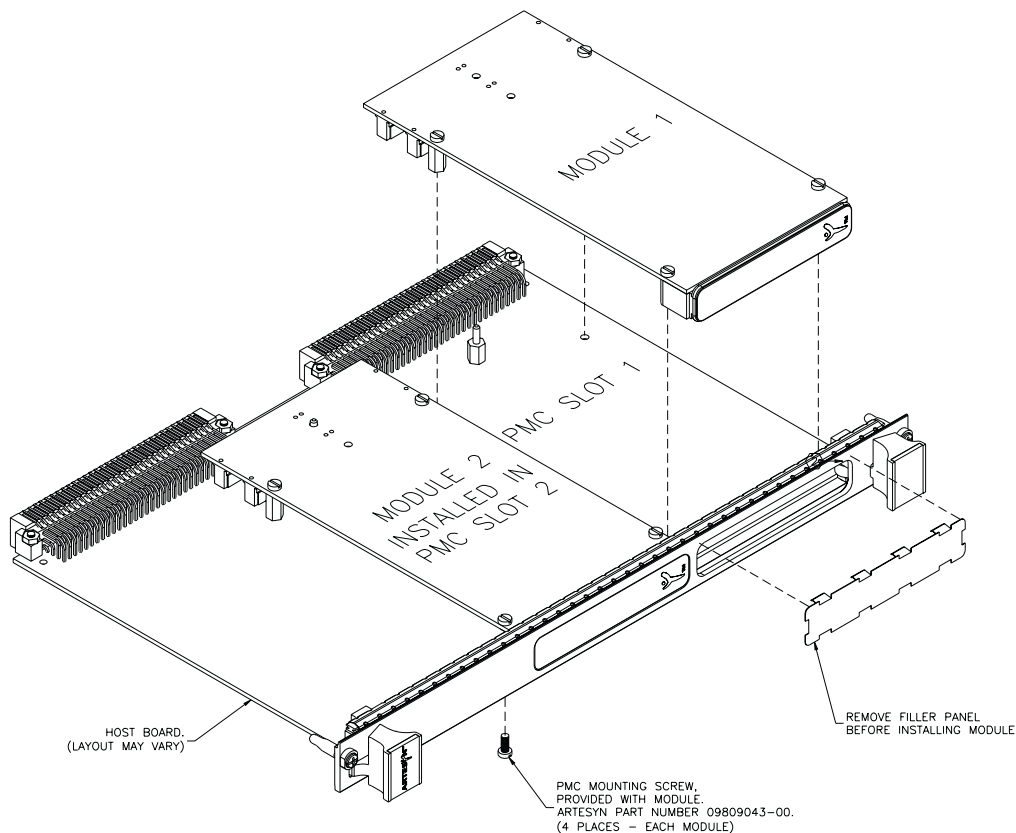


Figure 2-6. Installing the Module

4. Using four M2.5x6mm flathead screws, secure the PmPPC440 module from the bottom of the baseboard. First, insert and tighten the screws closest to the P11, P12, P13, and P14 connectors. Next, insert and tighten the screws nearest to the front panel.

2.4 Resets

The PmPPC440 has several sources for resets. The power-on sequence, the front panel reset switch, and a PCI reset all perform a hard reset to the entire board. Also, a software-controlled Reset Command register at C100,0008₁₆ allows for various types of resets, as follows:

3	2	1	0
PCI	Ethernet	Flash	Hard

Register Map 2-1. Configuration Command

- PCI

PCI Reset Command. Writing a one to this bit asserts the ResetOut signal on the PCI interface. The reset value is zero.
- Ethernet

Ethernet Reset Command. Writing a one to this bit causes an Ethernet reset. The reset value is zero.
- Flash

Flash Reset Command. Writing a one to this bit causes a Flash reset. The reset value is zero.
- Hard

Hard Reset Command. Writing a one to this bit causes a hard reset to the board. The reset value is zero.

After being written to, the Reset Command register bits clear automatically. The Hard Reset Command is disabled if the QS_QE bit is active (see Register Map 2-2). When the PCI Reset Command is active, it prevents a PCI reset from affecting the board.

The Quick Switch Enable register at C100,0034₁₆ provides for two reset modes. When bit zero is active (enabled), the PCI Reset signal connects to the PPC440GP processor’s SYS_RESET pin. When this bit is not active (disabled), the PmPPC440’s programmable logic device (PLD) drives the SYS_RESET pin. The reset value for the QS_QE bit is one (enabled).

3	2	1	0
reserved			QS_QE

Register Map 2-2. Quick Switch Enable

The Reset Event register at C100,0004₁₆ indicates the source of the last reset, as follows:

3	2	1	0
PCI	SW	FP	PUR

Register Map 2-3. Reset Event

- PCI** PCI Reset. A one in this bit indicates the reset was caused by a PCI event. The power-up value is zero.
- SW** Software Reset. A one in this bit indicates the reset was caused by a write to the PCI Command register. The power-up value is zero.
- FP** Front Panel Reset. A one in this bit indicates the reset was caused by the front panel push button. The power-up value is zero.
- PUR** Power-Up. A one in this bit indicates the reset was caused by the power-up sequence. The power-up value is one.

2.5 LED Control

The PmPPC440 has four, green, light-emitting diodes (LEDs) on the front panel to facilitate software development. The LED register at C100,0000₁₆ provides access to these LEDs. Writing a one to the appropriate bit illuminates the corresponding LED. The reset value is zero for all the LED bits.

3	2	1	0
LED4	LED3	LED2	LED1

Register Map 2-4. LED

2.6 Monarch Functionality

The PmPPC440 can function as either a Monarch or a non-Monarch module, as described in the VITA 32 PPMC draft specification. A Monarch is the main PPMC device on the local PCI bus. It performs enumeration on that bus after power-up and is often the interrupt handler. A non-Monarch module does not perform enumeration on the local bus after power-up. Bit 1 of the Board Configuration register (see Register Map 2-5) at location C100,0020₁₆ indicates how the module

is configured (1=Monarch, 0=non-Monarch), as determined by the signal on pin 64 of connector P12. The software can read the Monarch line status to configure the board without affecting the hardware.

The EReady register (see Register Map 5-1) at location C100,0010₁₆ has an additional bit to support Monarch functionality. Bit 0, ERdy, monitors the READY line. For a non-Monarch, the presumption is that this signal is initially asserted, then removed when the bus is ready for enumeration. When all the other PCI devices have stopped driving this signal low, the Monarch will enumerate the bus. Please see the draft PPMC standard (reference in Table 1-2) for carrier board pull-up requirements.

2.7 Board Configuration

The PmPPC440 has a register that provides configuration information about the board. The read-only Board Configuration register at C100,0020₁₆ identifies whether or not the board is a Monarch and whether or not it boots from the Development Mezzanine Card (DMC), as follows:

3	2	1	0
reserved		Monarch	DMC

Register Map 2-5. Board Configuration

- Monarch1 = Monarch, 0 = non-Monarch
- DMC1 = boot from PLCC on DMC, 0 = boot from PMC

To facilitate development, the PmPPC440 also has several read-only registers that contain identification and version information. The Product Identification register at C100,0014₁₆ holds a four-bit value that represents a unique identification number for the PmPPC440. The Hardware Version register at C100,0018₁₆ contains a hard-coded four-bit value that identifies the hardware version. The PLD Version register at C100,001C₁₆ contains a hard-coded four-bit value that identifies the programmable logic device’s code version.

2.8 Troubleshooting

In case of difficulty, use this checklist:

- ☐ Be sure the PmPPC440 circuit board is seated firmly in the card cage.
- ☐ Be sure the system is not overheating.
- ☐ Check the cables and connectors to be certain they are secure.
- ☐ If you are using the PmPPC440 monitor, run the power-up diagnostics and check the results. Chapter 8 describes the power-up diagnostics.
- ☐ Check your power supply for proper DC voltages. If possible, use an oscilloscope to look for excessive power supply ripple or noise (over 50 mV_{pp} below 10 MHz).
- ☐ Check that your terminal is connected to a console port.
- ☐ The PmPPC440 monitor uses values stored in on-card NVRAM (I²C EEPROM) to configure and set the baud rates for its console port. The lack of a prompt might be caused by incorrect terminal settings, incorrect configuration of the NVRAM, or a malfunctioning NVRAM.

2.8.1 Technical Support

After you have checked all of the above items, call 1-800-327-1251 and ask for technical support from our Product Support Services Department (or send e-mail to support@artesyincp.com). Please have the following information handy:

- PmPPC440 serial number
- baseboard model number and monitor revision level (if applicable)
- version and part number of the operating system (if applicable)
- board assembly/revision number from the sticker on the PmPPC440 board
- whether your board has been customized for options such as a higher processor speed or additional memory
- license agreements (if applicable)

2.8.2 Service Information

If you plan to return the board to Artesyn Communication Products for service, call 1-800-327-1251 and ask for our Test Services Department (or send e-mail to serviceinfo@artesyncp.com) to obtain a Return Merchandise Authorization (RMA) number. We will ask you to list which items you are returning and the board serial number, plus your purchase order number and billing information if your PmPPC440 hardware is out of warranty. Contact our Test Services Department for any warranty questions. If you return the board, be sure to enclose it in an antistatic bag, such as the one in which it was originally shipped. Send it prepaid to:

**Artesyn Communication Products
Test Services Department
8310 Excelsior Drive
Madison, WI 53717**

RMA # _____

Please put the RMA number on the outside of the package so we can handle your problem efficiently. Our service department cannot accept material received without an RMA number.

PPC440GP Processor

The PmPPC440 uses an IBM PPC440GP central processing unit (CPU), which has a 32-bit reduced instruction set computer (RISC) embedded processor. The processor runs at 400MHz and features debug facilities, timer facilities, and 32-kilo-byte instruction and data caches. The PPC440GP has a 128-bit processor local bus (PLB), a 32-bit on-chip peripheral bus (OPB), a 32-bit device control register (DCR) bus.

3.1 On-Chip Features

The PPC440GP integrates a number of system functions for the PmPPC440:

- Internal SRAM controller (ISC)
- Double data rate (DDR) synchronous DRAM (SDRAM) controller
- Peripheral component interconnect (PCI-X) bridge controller
- Direct memory access (DMA) controller
- Memory access layer (MAL) controller
- Support for two RMII PHYs
- Support for two on-chip Ethernet ports (EMACs)
- Two universal asynchronous receiver/transmitters (UARTs)
- Inter-integrated circuit (I2C) controller
- General purpose input/output (GPIO) interface

The following block diagram provides an overview of the PPC440GP architecture:

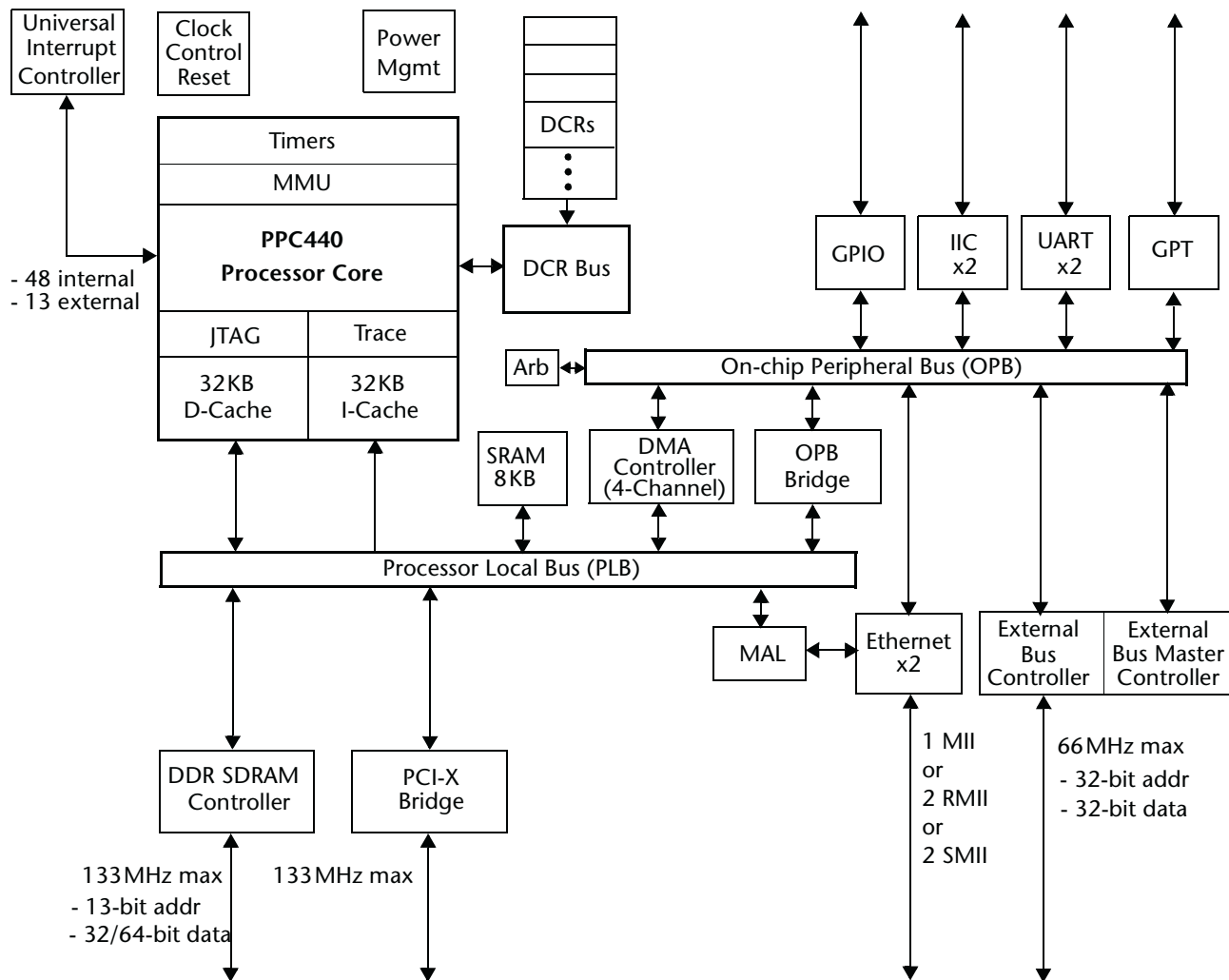


Figure 3-1. PPC440GP Block Diagram

3.2 Physical Memory Map

The PmPPC440 monitor (see Chapter 8) configures the memory map for the PPC440GP processor. The following figure shows the PmPPC440 physical memory map:

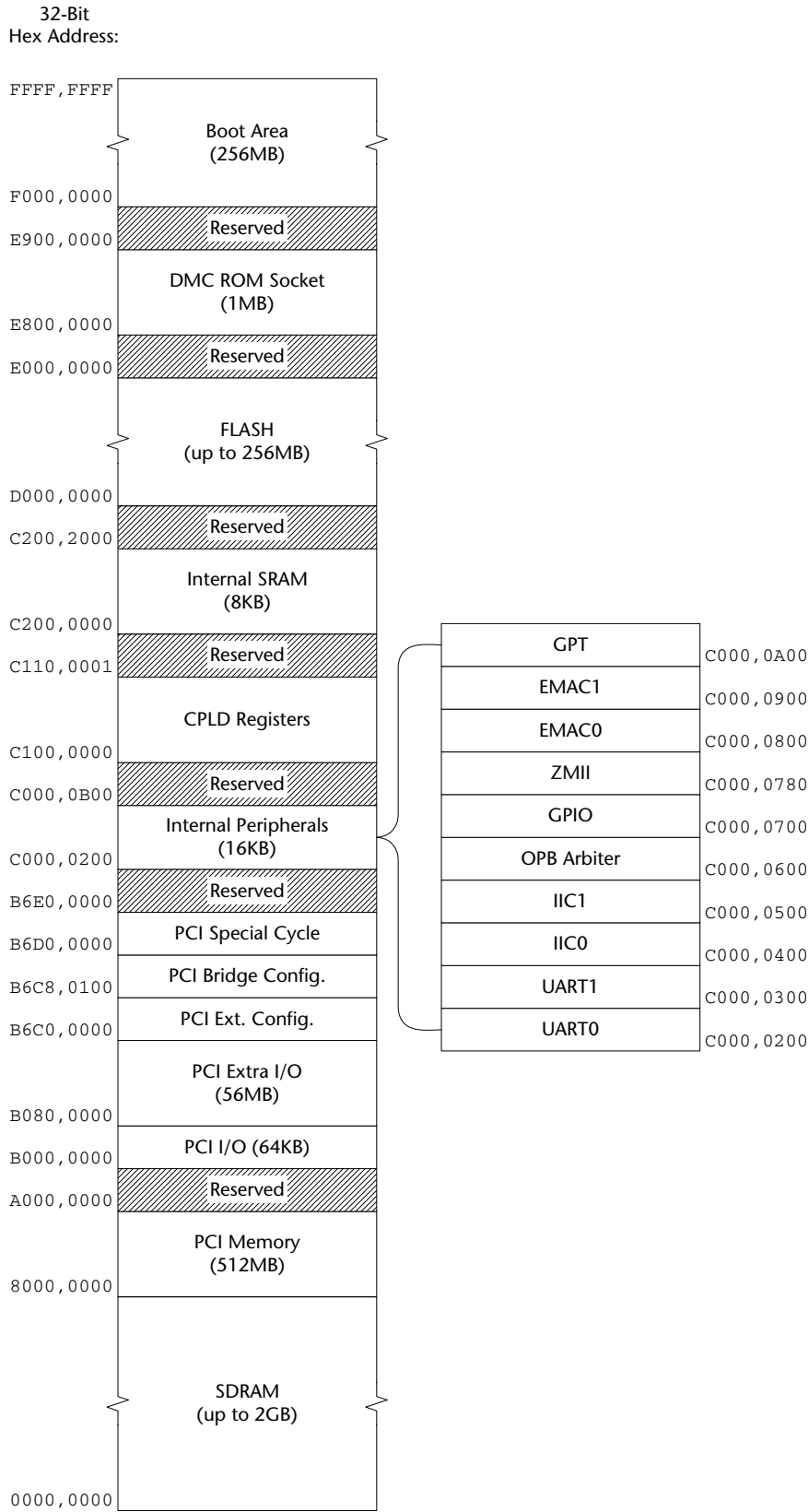


Figure 3-2. PPC440GP Memory Map

This table summarizes the physical addresses for the PPC440GP on the PmPPC440 board and provides a reference to more detailed information:

Table 3-1. PPC440GP Address Summary

Hex Address (32-bit)	Hex Address (36-bit)	Access Mode	Description	See Section
F000,0000	1,F000,0000	R	Boot Area (256MB)	8
E900,0000	1,E900,0000	—	Reserved	—
E800,0000	1,E800,0000	R	DMC ROM Socket (1MB)	7
E000,0000	1,E000,0000	—	Reserved	—
D000,0000	1,D000,0000	R/W	Flash (up to 256MB)	3.3.1
C200,2000	0,8000,2000	—	Reserved	—
C200,0000	0,8000,0000	R/W	SRAM (8KB)	3.1
C110,0001	1,C110,0001	—	Reserved	—
C100,0000	1,C100,0000	R/W	PLD Registers	3
C000,0B00	1,4000,0800	—	Reserved	—
C000,0200	1,4000,0200	R/W	Internal Peripherals (16KB)	8
B6E0,0000	2,0EE0,0000	—	Reserved	—
B000,0000	2,0800,0000	R/W	PCI I/O and Configuration	3
A000,0000	3,2000,0000	—	Reserved	—
8000,0000	3,0000,0000	R/W	PCI Memory Space	3
0000,0000	0,0000,0000	R/W	SO-DIMM SDRAM (up to 1GB)	3.3.2

3.3 On-Card Memory

The PmPPC440 has various types of on-card memory to support the PPC440GP. It has user Flash, SDRAM for data storage, and two serial EEPROMs for non-volatile memory storage. The following subsections describe these memory devices.

3.3.1 User Flash

The PmPPC440 user flash memory interface supports up to four 16-bit devices of 16, 32, or 64 megabytes each. This memory is arranged in four banks to support a maximum of 128 megabytes of contiguous memory for flash file systems. The flash is controlled by the PPC440GP and located at D000,0000₁₆ on the processor's peripheral bus (OPB).

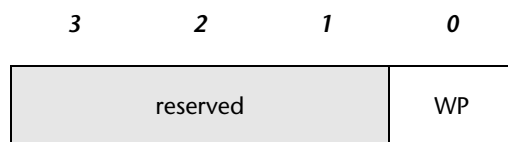
3.3.2 SDRAM

The PmPPC440 supports 16, 128, 256, and 512 megabytes of 72-bit wide synchronous dynamic random access memory (SDRAM). The PmPPC440 also will support one gigabyte of SDRAM (when the 1-gigabyte devices become available). The SDRAM interface implements eight additional bits to allow for error correcting code (ECC).

The SDRAM is in the form of a small-outline, dual in-line memory module (SO-DIMM) device. A serial EEPROM on the SO-DIMM provides configuration information, accessible via the I²C interface at address A2₁₆. The SDRAM occupies physical addresses from 0000,0000₁₆ to 7FFF,FFFF₁₆ on the Processor Local Bus (PLB). The PPC440GP processor controls the SDRAM and supports a double data rate (DDR) interface that allows for transfer speeds of up to 266MHz.

3.3.3 EEPROMs

The PPC440GP uses a two-kilobit read-only serial EEPROM at hex location A0₁₆ on the I²C bus to store PPC440GP configuration data. If the configuration data becomes corrupted, the processor may hang after reset. To reduce the risk of corruption, the Configuration Write-Protect register at C100,0030₁₆ controls access to the write-protect (WP) pin of the EEPROM. For bit zero: 1 = write-protected (default), 0 = not write-protected.



Register Map 3-1. Configuration Write-Protect

The PmPPC440 also has a 64-kilobit serial EEPROM at hex location A8₁₆ on the I²C bus to provide additional non-volatile memory space.

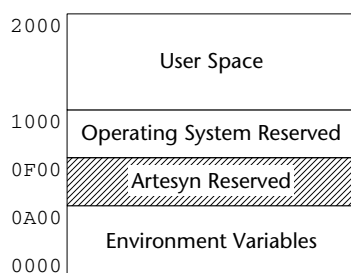


Figure 3-3. Memory Map for 64-Kilobit EEPROM

3.4 Serial I/O

The PPC440GP has internal universal asynchronous receiver/transmitters (UARTs) to support two EIA-232 serial ports. Dual serial ports can be routed to the P14 PMC connector (see Table 5-2 for pinouts). Alternatively, a single serial port can be routed to the front panel mini-B Universal Serial Bus (USB) connector, P2. The following table lists the pinouts for P2:

Table 3-2. USB Serial Port Pin Assignments, P2

Pin	Signal
1	no connection
2	RX
3	TX
4	no connection
5	ground

3.5 Ethernet

The PPC440GP provides two reduced media independent interface (RMII) connections to support two optional 10/100BaseTX Ethernet ports on the PmPPC440 front panel (see Table 6-2 for pin assignments). The optional Ethernet signals also route to the P14 PMC connector (see Table 5-2 for pin assignments). Please refer to Chapter 6 for additional information about the Ethernet interface.

3.6 I²C Interface

The PPC440GP has a built-in inter-integrated circuit (I²C) interface that supports master and slave I²C devices. The following devices connect to the I²C bus:

- 2-kilobit serial EEPROM for PPC440GP configuration data
- 64-kilobit serial EEPROM
- real-time clock (RTC) device
- SO-DIMM serial presence detection (SPD)
- P14 PMC connector

3.7 Miscellaneous Status Register

The PmPPC440 has a read-only Miscellaneous Status register at C100,002C₁₆ that allows software to monitor the status of the wait signal from the flash and the system error signal from the PPC440GP. Bit one monitors the GP440_SYSERR signal, and bit zero monitors the FLASH_WAIT signal.

Real-Time Clock

The standard real-time clock (RTC) for the PmPPC440 is provided by an M41T00 device from STMicroelectronics. This device has an integrated year-2000-compatible RTC, power sense circuitry, and uses eight bytes of non-volatile RAM for the clock/calendar function. The M41T00 is powered from the +3.3 V rail during normal operation.

CAUTION. A supercap on the PmPPC440 provides backup operation in the event of a power failure. However, if power is not reapplied within 12 hours, all data stored in non-volatile RAM may be lost.

4.1 Block Diagram

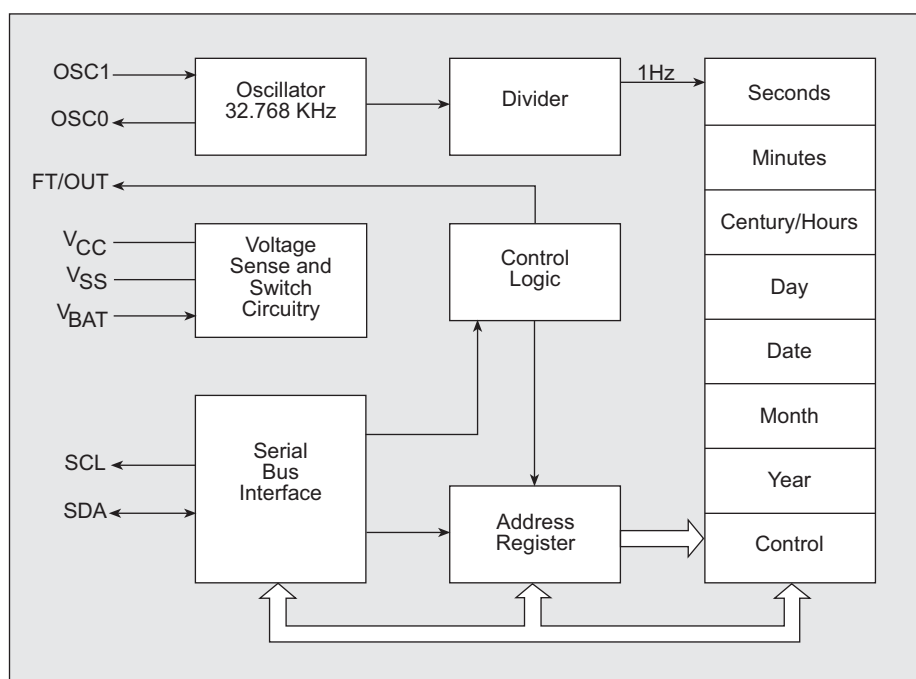


Figure 4-1. M41T00 Real-Time Clock Block Diagram

4.2 Operation

The M41T00 clock operates as a slave device on the serial bus. To obtain access, the RTC implements a start condition followed by the correct slave address (DO_{16}). Access the eight bytes in the following order:

1. Seconds register
2. Minutes register
3. Century/Hours register
4. Day register
5. Date register
6. Month register
7. Years register
8. Control register

The M41T00 clock continually monitors the supply voltage (V_{cc}) for an out of tolerance condition. If V_{cc} falls below switch-over voltage (V_{so}), the M41T00:

- Terminates an access in progress
- Resets the device address counter
- Does not recognize inputs (prevents erroneous data from being written)

At power-up, the M41T00 uses V_{cc} at V_{so} and recognizes inputs.

4.3 Clock Operation

Read the seven Clock registers one byte at a time or in a sequential block. Access the Control register (address location 7) independently. An update to the Clock registers is delayed for 250 ms to allow the read to be completed before the update occurs. This delay does not alter the actual clock time. The eight byte clock register sets the clock and reads the date and time from the clock, as summarized in Table 4-1:

Table 4-1. RTC Register Map

Address	Data								Function/Range	
	D7	D6	D5	D4	D3	D2	D1	D0		
00	ST	10 Seconds			Seconds			Seconds	00–59	
01	X	10 Minutes			Minutes			Minutes	00–59	
02	CEB	CB	10 Hours		Hours			Century/Hours	0-1/00-23	
03	X	X	X	X	X	Day		Day	01–07	
04	X	X	10 Date		Date			Date	01–31	
05	X	X	X	10 M	Month			Month	01–12	
06	10 Years				Years			Years	00–99	
07	OUT	FT	S	Calibration				Control	–	

ST Stop bit.
 1=Stops the oscillator
 0=Restarts the oscillator within one second

CEB Century Enable Bit.
 1=Causes CB to toggle either from 0 to 1 or from 1 to 0 at the turn of the century
 0=CB will not toggle

CB Century Bit.

Day Day of the week.

Date Day of the month.

OUT Output level.
 1=Default at initial power-up
 0=FT/OUT (pin 7) driven low when FT is also zero

FT Frequency Test bit.
 1=When oscillator is running at 32,768 Hz, the FT/OUT pin will toggle at 512 Hz
 0=The FT/OUT pin is an output driver (default at initial power-up)

S Sign bit.
 1=Positive calibration
 0=Negative calibration

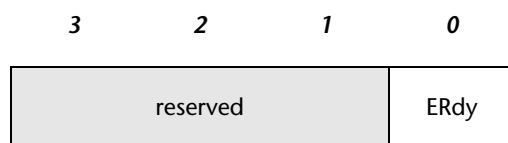
- Calibration** Calibration bits. The calibration circuit adds or subtracts counts from the oscillator divider circuit at the divide by 256 stage. The number of times pulses are blanked (subtracted, negative calibration) or split (added, positive calibration) depends on this five-bit byte. Adding counts accelerates the clock, and subtracting counts slows the clock down.
- X Don't care bit.

PCI/PCI-X Interface

The Artesyn PmPPC440 module complies with the Peripheral Component Interconnect (PCI) bus interface standard and the associated PCI Mezzanine Card (PMC) mechanical interface standard. The PmPPC440 also supports the faster bus speed of 133MHz, as described in the PCI-X v1.0 specification. The PmPPC440 requires a PMC/PCI-compliant baseboard.

5.1 PCI Enumeration

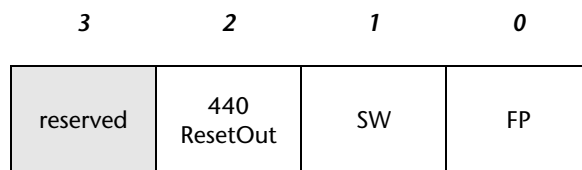
The EReady register at C100,0010₆ provides for PCI enumeration status and control. If the PmPPC440 is a Monarch, the register is read only (1 = ready for enumeration, 0 = not ready for enumeration). If it is not a Monarch, the register is writable (0 = PMC not ready for enumeration [default], 1 = PMC is ready for enumeration).



Register Map 5-1. EReady

5.2 PCI Reset Control

The PMC Reset Enable register at C100,000C₁₆ determines the functionality of the PCI ResetOut signal, as follows:



Register Map 5-2. PMC Reset Enable

440 ResetOut	CPU Reset. This bit allows the PCI ResetOut signal to be driven when the PPC440GP ResetOut signal is driven (1=enable, 0=disable). The reset value is zero.
SW	Software Reset. This bit allows the PCI ResetOut signal to be driven when an on-board hard reset is caused by a write to the Reset Command register (1=enable, 0=disable). The reset value is zero.
FP	Front Panel Reset. This bit allows the PCI ResetOut signal to be driven when an on-board hard reset is caused by the front panel push button (1=enable, 0=disable). The reset value is one.

5.3 PCI Bus Control Signals

The following signals for the PCI interface are available on connectors P11–P13. Refer to the PCI specification for details on using these signals. All signals are bi-directional unless otherwise stated.

NOTE. A sustained three-state line is driven high for one clock cycle before float.

ACK64*, REQ64*	These output signals are used to tell a 64-bit PCI device whether to use the 64-bit or the 32-bit data width. Since the PmPPC440 is a 64-bit board, these signals are tied off to indicate the 64-bit data width.
AD00-AD31	ADDRESS and DATA bus (bits 0-31). These three-state lines are used for both address and data handling. A bus transaction consists of an address phase followed by one or more data phases.
C/BE0*-C/BE3*	BUS COMMAND and BYTE ENABLES. These three-state lines have different functions depending on the phase of a transaction. During the address phase of a transaction these lines define the bus command. During a data phase the lines are used as byte enables.
CLK	CLOCK. This input signal to the PmPPC440 provides timing for PCI transactions.
DEVSEL*	DEVICE SELECT. This sustained three-state signal indicates when a device on the bus has been selected as the target of the current access.
EREDDY	READY. This signal is an input for Monarch modules and an output for non-Monarch modules. It indicates that all modules are initialized and the PCI bus is ready to be enumerated.
FRAME*	CYCLE FRAME. This sustained three-state line is driven by the current master to indicate the beginning of an access, and continues to be asserted until transaction reaches its final data phase.

GNT*	GRANT. This input signal indicates that access to the bus has been granted to a particular master. Each master has its own GNT*.
IDSEL	INITIALIZATION DEVICE SELECT. This input signal acts as a chip select during configuration read and write transactions.
INTA*, INTB*, INTC*, INTD*	PMC INTERRUPTS A, B, C, D. These interrupt lines are used by PCI devices to interrupt the host processor.
IRDY*	INITIATOR READY. This sustained three-state signal indicates that the bus master is ready to complete the data phase of the transaction.
M66EN	ENABLE 66MHZ. When grounded, this signal prevents 66MHz operation of the PCI bus.
MONARCH*	MONARCH. When this signal is grounded, it indicates that the PmPPC440 is a Monarch and must provide PCI bus enumeration and interrupt handling.
LOCK*	LOCK. This sustained three-state signal indicates that an automatic operation may require multiple transactions to complete.
PAR	PARITY. This is even parity across AD00-AD31 and C/BE0-C/BE3*. Parity generation is required by all PCI agents. This three-state signal is stable and valid one clock after the address phase, and one clock after the bus master indicates that it is ready to complete the data phase (either IRDY* or TRDY* is asserted). Once PAR is asserted, it remains valid until one clock after the completion of the current data phase.
PCIXCAP	PCI-X CAPABILITY. This line indicates which type of PCI/PCI-X card is attached to the PCI/PCI-X bus. If it is pulled up to 3.3V, the bus operates in 133-MHz PCI-X mode. If the line is grounded, the bus operates in conventional PCI mode. If the line is connected to a voltage divider, the bus operates in 66-MHz PCI-X mode.
PERR*	PARITY ERROR. This sustained three-state line is used to report parity errors during all PCI transactions.
PME*	POWER MANAGEMENT EVENT. This optional open-drain signal (pull-up resistor required) allows a device to request a change in the power state. Devices must be enabled by software before asserting this signal.
PRESENT*	PRESENT. When grounded, this signal indicates to a carrier that a PMC module is installed.
RESET_OUT*	RESET OUTPUT. This optional output signal may be used to support another source. To avoid reset loops, do not use RST* to generate RESET_OUT*.
REQ64*	This optional output signal is used to tell a 64-bit PCI device whether to use the 64-bit or the 32-bit data width.

REQ*	REQUEST. This output pin indicates to the arbiter that a particular master wants to use the bus.
RST*	RESET. The assertion of this input line brings PCI registers, sequencers, and signals to a consistent state.
SERR*	SYSTEMS ERROR. This open-collector output signal is used to report any system error with catastrophic results.
STOP*	STOP. This is a sustained three-state signal used by the current target to request that the bus master stop the current transaction.
TRDY*	TARGET READY. This is a sustained three-state signal that indicates the target's ability to complete the current data phase of the transaction.

5.4 Additional PMC Signals

The following signals route to the PCI expansion connector at P14. (Refer to Table 5-2 for pinouts.)

440GP I²C SDA	This is a TTL-level input/output signal for I ² C serial data.
440GP I²C SCL	This is a TTL-level input signal to the PMC for the I ² C serial clock.
Serial1 TxData, Serial2 TxData	This is the transmit data output signal (TTL or EIA-232) from the PMC for Serial Ports 1 and 2, respectively.
Serial1 RxData, Serial2 RxData	This is the receive data input signal (TTL or EIA-232) to the PMC for Serial Ports 1 and 2, respectively.
GPIO0–GPIO3	These are TTL-level input/output signals for general-purpose I/O. They connect to the PPC440GP's GPIO ports.
Ethernet1 TD_P, Ethernet2 TD_P	This is the positive side of the differential analog transmit data signal for Ethernet Ports 1 and 2, respectively.
Ethernet1 TD_N, Ethernet2 TD_N	This is the negative side of the differential analog transmit data signal for Ethernet Ports 1 and 2, respectively.
Ethernet1 RD_P, Ethernet2 RD_P	This is the positive side of the differential analog receive data signal for Ethernet Ports 1 and 2, respectively.
Ethernet1 RD_N, Ethernet2 RD_N	This is the negative side of the differential analog receive data signal for Ethernet Ports 1 and 2, respectively.

5.5 PMC Connector Pin Assignments

The PmPPC440 has four 64-pin PMC connectors. Table 5-1 shows the pin assignments for P11, P12, and P13.

Table 5-1. PMC Connector Pin Assignments, P11–P13

Pin	P11	P12	P13	Pin	P11	P12	P13
1	ISP_TCK	no connection	no connection	33	FRAME*	Ground	GND
2	no connection	ISP_TRST*	GND	34	Ground	no connection	AD48
3	Ground	ISP_TMS	GND	35	Ground	TRDY*	AD47
4	INTA*	ISP_TDO	C/BE7*	36	IRDY*	+3.3V	AD46
5	INTB*	ISP_TDI	C/BE6*	37	DEVSEL*	Ground	AD45
6	INTC*	Ground	C/BE5*	38	+5V (unused)	STOP*	GND
7	PRESENT*	Ground	C/BE4*	39	PCIXCAP	PERR*	V(I/O)
8	+5V (unused)	no connection	GND	40	no connection	Ground	AD44
9	INTD*	no connection	V(I/O)	41	no connection	+3.3V	AD43
10	no connection	no connection	PAR64	42	no connection	SERR*	AD42
11	Ground	PUP	AD63	43	PAR	C/BE1*	AD41
12	no connection	+3.3V	AD62	44	Ground	Ground	GND
13	PCICLK	RST*	AD61	45	V(I/O)	AD14	GND
14	Ground	PDN	GND	46	AD15	AD13	AD40
15	Ground	+3.3V	GND	47	AD12	M66EN	AD39
16	GNT*	PDN	AD60	48	AD11	AD10	AD38
17	REQ*	no connection	AD59	49	AD09	AD08	AD37
18	+5V (unused)	Ground	AD58	50	+5V (unused)	+3.3V	GND
19	V(I/O)	AD30	AD57	51	Ground	AD07	GND
20	AD31	AD29	GND	52	C/BE0*	no connection	AD36
21	AD28	Ground	V(I/O)	53	AD06	+3.3V	AD35
22	AD27	AD26	AD56	54	AD05	no connection	AD34
23	AD25	AD24	AD55	55	AD04	no connection	AD33
24	Ground	+3.3V	AD54	56	Ground	Ground	GND
25	Ground	IDSEL	AD53	57	V(I/O)	no connection	V(I/O)
26	C/BE3*	AD23	GND	58	AD03	EREADY*	AD32
27	AD22	+3.3V	GND	59	AD02	Ground	no connection
28	AD21	AD20	AD52	60	AD01	RESETOUT*	no connection
29	AD19	AD18	AD51	61	AD00	ACK64*	no connection
30	+5V (unused)	Ground	AD50	62	+5V (unused)	+3.3V	GND
31	V(I/O)	AD16	AD49	63	Ground	Ground	GND
32	AD17	C/BE2*	GND	64	REQ64*	MONARCH*	no connection

Table 5-2 shows the pin assignments for P14.

Table 5-2. PMC Connector Pin Assignments, P14

Pin	P14	Pin	P14
1	440GP I ² C SDA	33	no connection
2	440GP I ² C SCL	34	no connection
3	Serial1 TxData	35	no connection
4	Serial1 RxData	36	no connection
5	Serial2 TxData	37	no connection
6	Serial2 RxData	38	no connection
7	GPIO0	39	no connection
8	Ground	40	no connection
9	Ground	41	no connection
10	GPIO1	42	no connection
11	GPIO2	43	no connection
12	GPIO3	44	no connection
13	Ethernet1 TD_P	45	no connection
14	Ethernet1 TD_N	46	no connection
15	Ethernet1 RD_P	47	no connection
16	Ethernet1 RD_N	48	no connection
17	Ethernet2 TD_P	49	no connection
18	Ethernet2 TD_N	50	no connection
19	Ethernet2 RD_P	51	no connection
20	Ethernet2 RD_N	52	no connection
21-32	no connection	53-64	no connection

Ethernet Interface

The PPC440GP processor provides an Ethernet interface that supports two 10/100BaseTX Ethernet ports for the PmPPC440. The interface uses two Broadcom BCM5221 PHY transceivers to provide two isolated Ethernet ports to the front panel or two non-isolated ports to the P14 PMC connector.

6.1 BCM5221 Registers

The MII management interface registers are serially written to and read from using the management data I/O (MDIO) and management data clock (MDC) pins. The MII register map summary is specified in Table 6-1. When writing to the reserved bits, ignore the output value. The initialization column is the reset value of the register.

Table 6-1. BCM5221 MII Register Map Summary

Hex Address	Name	Initializa- tion (Hex)
00	Control	3000
01	Status	782D
02	PHYID High	0040
03	PHYID Low	61E4
04	Auto-Negotiation Advertisement	01E1
05	Auto-Negotiation Link Partner Ability	0021
06	Auto-Negotiation Expansion	0004
07	Auto-Negotiation Next Page	2001
08	Auto-Negotiation Link Partner Next Page Transmit	0000
10	100BASE-X Auxiliary Control	0000
11	100BASE-X Auxiliary Status	0001
12	100BASE-X Receive Error Counter	0000
13	100BASE-X False Carrier Sense Counter	0000
14	100BASE-X Disconnect Counter	0200

Table 6-1. BCM5221 MII Register Map Summary — *Continued*

Hex Address	Name	Initializa- tion (Hex)
15	Reserved	0600
16	Reserved	0100
17	PTest	0000
18	Auxiliary Control/Status	003x
19	Auxiliary Status Summary	0006
1A	Interrupt	9F0x
1B	Auxiliary Mode2	008A
1C	10BASE-T Auxiliary Error and General Status	082x
1D	Auxiliary Mode	0000

6.2 Ethernet Addresses

The Ethernet address for your board is a unique identifier on a network and must not be altered. The address consists of 48-bits divided into two equal parts. The upper 24-bits define a unique identifier that has been assigned to Artesyn Communication Products, Inc. by IEEE. The lower 24-bits are defined by Artesyn for identification of each of our products.

The Ethernet address for the PmPPC440 is a binary number referenced as 12 hexadecimal digits separated into pairs, with each pair representing 8-bits. The address assigned to the PmPPC440 has the following form:

**00 80 F9 69 0X XX or
00 80 F9 69 8X XX**

00 80 F9 is Artesyn's identifier. **69** is the identifier for the PmPPC440 product group. The last two pairs of hex numbers correspond to the following formula: $n - 1000$, where n is the unique serial number assigned to each board. However, in order to differentiate between the two Ethernet ports on the board, the fifth pair of hex numbers always begins with **0** for port A and **8** for port B.

For example, if the serial number of a PmPPC440 is 2867, the calculated value is 1867 ($74B_{16}$). Therefore, the board's Ethernet address for port A is 00:80:F9:58:07:4B, and the address for port B is 00:80:F9:58:87:4B.

6.3 Front Panel Ethernet Connectors

The PmPPC440 has two, optional, RJ45, Ethernet connectors located on the front panel. The pinouts for these connectors are as follows:

Table 6-2. Ethernet Pin Assignments, P3 & P4

Pin	Signal	Pin	Signal
1	TX+	6	RX–
2	TX–	7	common
3	RX+	8	common
4	common	9	no connection
5	common	10	no connection

Development Mezzanine Card

The development mezzanine card (DMC) is an optional plug-on card mounted on the front of the PmPPC440 board to expedite product development. This chapter describes the physical layout of the DMC, the setup process, and how to check for proper operation once the board has been installed. The DMC facilitates hardware and software development by providing:

- Four LEDs for software development (connected to the PPC440GP GPIO pins)
- A COP header for software development
- JTAG header for CPLD programming
- A 32-pin PLCC 8-pin ROM socket for software development
- Four software-readable jumpers for development use

NOTE. The DMC has dual RJ45 connectors to support two 10/100BASE-TX Ethernet ports and a USB connector to support an EIA-232 serial debug port. However, these features are not used in conjunction with the PmPPC440 product.

7.1 DMC Circuit Board

The DMC is a custom four-layer circuit board. It has the following physical dimensions:

Table 7-1. DMC Mechanical Specifications

Width	Depth	Height, Top Side	Height, Bottom Side
2.913 in. (74.0 mm)	2.100 in. (53.3 mm)	0.323 in. (8.2 mm)	0.007 in. (1.9 mm)

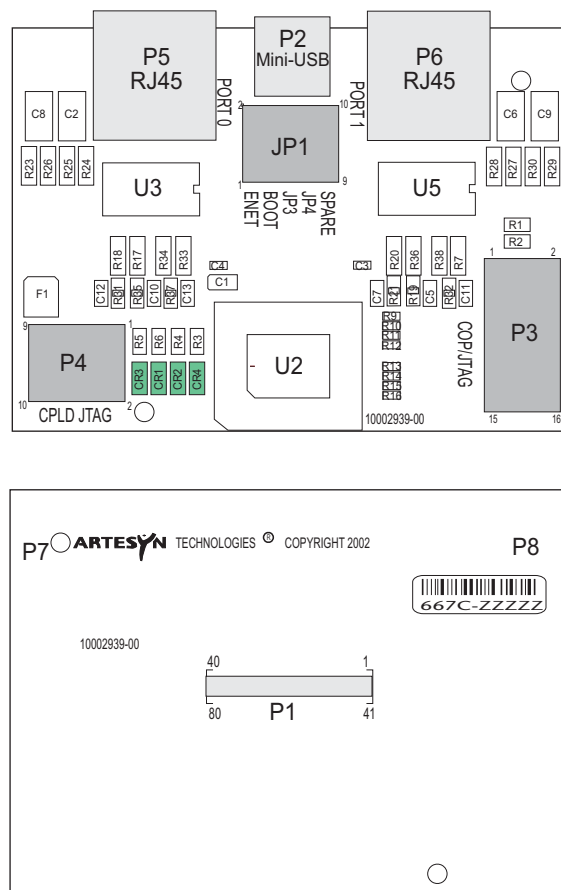


Figure 7-1. DMC Component Maps, Top and Bottom (Rev. 01)

7.1.1 Serial Numbers

Before you install the DMC in a system, you should record the following information:

- ☐ The board serial number: 667C-_____.

The board serial number appears on a bar code sticker located on the bottom of the board.

- ☐ The board assembly/configuration description: _____.

A sticker on the board contains the board assembly part number and configuration description. Be sure to include all the information that appears on the sticker.

- Any custom or user ROM installed, including version and serial number:

_____ .

It is useful to have these numbers available when you contact Technical Support or Test Services at Artesyn Communication Products.

7.2 Connectors

The DMC has the following connectors

- P1** P1 is an 80-pin PCB-to-PCB female connector on the bottom side of the DMC. This routes memory, CPLD, and CPU signals from the PmPPC440 to the DMC for development use. See Table 7-2 for the pin assignments.
- P2** P2 is a mini-B USB 9-pin connector. This provides the EIA-232 interface (unused by the PmPPC440). See Table 7-3 for the pin assignments.
- P3** The 16-pin COP/JTAG interface header allows software development to the PPC440GP. Refer to Table 7-4 for the pin assignments.
- P4** The CPLD JTAG header provides access to the CPLD programming interface. Refer to Table 7-5 for the pin assignments.
- P5-P6** P5 (Port 0) and P6 (Port 1) are the 10/100 fast Ethernet RJ45 connectors (unused by the PmPPC440). Refer to Table 7-6 for the pin assignments.

7.2.1 DMC Connector Pin Assignments

Connector P1 is an 80-pin connector that routes memory, CPLD, CPU, and Ethernet signals from the PmPPC440 to the DMC. Table 7-2 shows the pin assignments.

Table 7-2. DMC Connector Pin Assignments, P1

Pin	Signal	Pin	Signal
1	3.3V	41	3.3V
2	CPLD_TCK	42	CPU_TCK
3	GND	43	GND
4	no connection	44	no connection
5	no connection	45	no connection
6	CS*	46	no connection
7	OE*	47	no connection
8	WR*	48	no connection
9	LA17	49	no connection

Table 7-2. DMC Connector Pin Assignments, P1 — *Continued*

Pin	Signal	Pin	Signal
10	LA16	50	no connection
11	LA15	51	JP1
12	LA14	52	BOOT_SRC
13	LA13	53	JP3
14	LA12	54	JP4
15	LA11	55	LED1*
16	LA10	56	LED2*
17	LA9	57	LED3*
18	LA8	58	LED4*
19	LA7	59	CPU_TDO
20	LA6	60	CPU_TDI
21	LA5	61	CPU_TRST*
22	LA4	62	CPU_TMS
23	LA3	63	no connection
24	LA2	64	no connection
25	BADDR2	65	no connection
26	BADDR1	66	no connection
27	BADDR0	67	no connection
28	AD7	68	no connection
29	AD6	69	no connection
30	AD5	70	no connection
31	AD4	71	no connection
32	AD3	72	CPLD_TDI
33	AD2	73	CPLD_TMS
34	AD1	74	CPLD_TDO
35	AD0	75	DMC_DETECT
36	SIO1_TX	76	no connection
37	SIO1_RX	77	no connection
38	GND	78	GND
39	CPU_VIO	79	no connection
40	3.3V	80	3.3V

3.3 V	3.3 V is the power supply to the DMC (analog).
CPLD_TCK	PLD Test Clock is an input to DMC and part of the PLD JTAG interface.
CS*	Chip Select for DMC Flash is an input to DMC.
OE*	Output Enable for DMC Flash is an input to DMC.
WR*	Write Enable for DMC Flash is an input to DMC.
LA(17:2)	Latched Address for DMC Flash is an input to DMC.
BADDR(2:0)	Burst Address for DMC Flash is an input to DMC.
AD(7:0)	Multiplexed Address/Data bus for DMC Flash data is an output from DMC.
SIO1_TX	Serial IO Transmit (console port) is an input to DMC.
SIO1_RX	Serial IO Receive (console port) is an output from DMC.
CPU_VIO	IO Voltage for CPU is used as reference/power on the debug header (analog).
CPU_TCK	CPU Test Clock is an output from DMC and part of CPU JTAG interface.
BOOT_SRC	Boot source is an output from DMC and indicates to the PmPPC440 whether to boot from the DMC socketed Flash or the PmPPC440 soldered Flash.
JP(4:1)	Jumper signals are an output from the DMC.
LED(4:1)*	These DMC LEDs are an input to DMC and are user definable for development purposes.
CPU_TDO	CPU Test Data Out is an input to DMC and part of CPU JTAG interface.
CPU_TDI	CPU Test Data In is an output from DMC and part of CPU JTAG interface.
CPU_TRST*	CPU Test Reset is an output from DMC and part of CPU JTAG interface.
CPU_TMS	CPU Test Mode Select is an output from DMC and part of CPU JTAG interface.
CPLD_TDI	PLD Test Data In is part of the PLD JTAG interface (analog).

CPLD_TMS	PLD Test Mode Select is an output from DMC and part of PLD JTAG interface.
CPLD_TDO	PLD Test Data Out is an input to DMC and part of PLD JTAG interface.
DMC_DETECT	DMC Presence Detect is an output from DMC and indicates to the PMC that the DMC is installed.

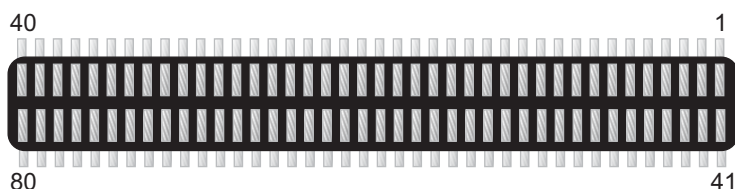


Figure 7-2. DMC PCB-To-PCB Connector, P1

7.2.2 EIA-232 Interface (Unused)

Connector P2 is a mini-B, Universal Serial Bus (USB), 9-pin connector for the EIA-232 interface.

Table 7-3. DMC USB Connector Pin Assignments, P2

Pin	Signal	Pin	Signal
1	no connection	4	no connection
2	DMC_RXD (Input)	5	GND
3	DMC_TXD (Output)	6-9	Connector housing ground

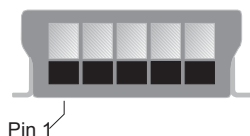


Figure 7-3. DMC Mini-B USB Connector, P2

7.2.3 JTAG/COP Interface

The JTAG/COP interface provides for boundary-scan testing of the CPU and the PmPPC440. This interface is compliant with the IEEE 1149.1 standard.

Table 7-4. DMC JTAG/COP Connector Pin Assignments, P3

Pin	Signal	Pin	Signal
1	TDO	9	TMS
2	no connection	10	no connection
3	TDI	11	no connection
4	TRST	12	no connection
5	no connection	13	no connection
6	VDD_SENSE	14	Key ^a
7	TCK	15	no connection
8	no connection	16	GND
^a Pin 14 is not installed.			

- TCK** Test Clock Input. Scan data is latched at the rising edge of this signal.
- TDI** Test Data Input. This signal acts as the input port for scan instructions and data.
- TDO** Test Data Output. This signal acts as the output port for scan instructions and data.
- TMS** Test Mode Select. This input signal is the test access port (TAP) controller mode signal.
- TRST** Test Reset. This input signal resets the test access port.

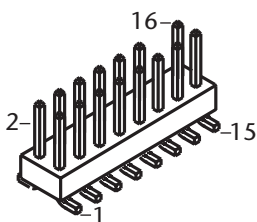


Figure 7-4. DMC JTAG/COP Header, P3

7.2.4 JTAG Chain Header

This header allows access to the CPLD programming interface.

Table 7-5. DMC JTAG Chain Header Pin Assignments, P4

Pin	Signal	Pin	Signal
1	TCK	6	no connection
2	GND	7	no connection
3	TDO	8	no connection
4	Fused 3.3 V	9	TDI
5	TMS	10	GND

TCK Test Clock Input. This is the clock input to the boundary scan test (BST) circuitry. Some operations occur at the rising edge, while others occur at the falling edge.

TDI Test Data Input. This is the serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK.

TDO Test Data Output. This is the serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK.

TMS Test Mode Select. This input pin provides the control signal to determine the transitions of the TAP controller state machine. Transitions within the state machine occur at the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK.

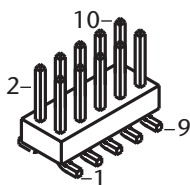


Figure 7-5. DMC JTAG Chain Header, P4

7.2.5 Ethernet Ports (Unused)

P5 and P6 are RJ45 connectors that allow for 10/100BASE-T Fast Ethernet access.

Table 7-6. DMC Ethernet Port Pin Assignments, P5–P6

Pin	Signal	Pin	Signal
1	TD_P	7	no connection
2	TD_N	8	no connection
3	RD_P	9	no connection
4	no connection	10	no connection
5	no connection	11	GND
6	RD_N	12	GND

- TD_P** Transmit Data Positive side of differential signal.
- TD_N** Transmit Data Negative side of differential signal.
- RD_P** Receive Data Positive side of differential signal.
- RD_N** Receive Data Negative side of differential signal.

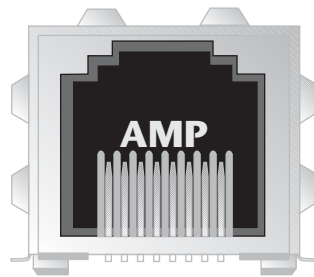


Figure 7-6. DMC RJ45 Connectors, P5-P6

7.3 DMC Jumpers

There are a total of five jumper pairs on the DMC. Pins 9 and 10 are spare jumper posts. See Fig. 7-1 for the jumper location on the DMC.

- JP1** This is a user-defined jumper.
- JP2** JP2 (pins 3 and 4) selects the 8-bit ROM socket as the boot device. In order for the socket to provide boot code, the DMC must be seated on the PmPPC440 and the boot jumper must be in place.
- JP3** This is a user-defined jumper.
- JP4** JP4 is the PPC440GP serial ROM configuration jumper. If JP4 is installed, the PPC440GP will not try to configure from the serial ROM.

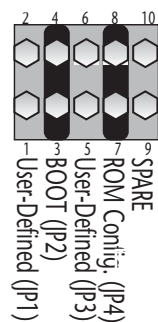


Figure 7-7. DMC Jumper Pin Assignments, JP1

7.3.1 Jumper Setting Register

These read-only bits may be read by software at location C100,0028₁₆ to determine the current DMC jumper (JP1) settings.

7	6	5	4	3	2	1	0
Reserved				JP4	JP3	JP2	JP1

Register Map 7-1. DMC Jumper

- JP4** Jumper 4 on DMC (PPC440GP serial ROM configuration):
1 = Installed (PPC440GP will not configure from ROM)
0 = Not installed (PPC440GP will configure from ROM)
- JP3** Jumper 3 on DMC (user defined):
1 = Installed
0 = Not installed
- JP2** Jumper 2 on DMC (BOOT):
1 = Installed (Boot from DMC ROM socket)
0 = Not installed (Boot from PmPPC440 Flash)
- JP1** Jumper 1 on DMC (user defined):
1 = Installed
0 = Not installed

7.4 Debug/Status LEDs

The DMC has four green, light-emitting diodes (LEDs) for software development; see Fig. 7-1 for LED locations. These LEDs are controlled by the PPC440GP through its general-purpose input/output bits, GPIO[31:28].

7.5 DMC Setup

You need the following items to set up and check the operation of the Artesyn DMC.

- ☐ A compatible PPMC board, such as the Artesyn PmPPC440
- ☐ Card cage and power supply
- ☐ CRT terminal

When you unpack the board, save the antistatic bag and box for future shipping or storage.

CAUTION. Do not install the board in a rack or remove the board from a rack while power is applied, at risk of damage to the board.

7.5.1 Installing the DMC Card

Use the following procedure to attach the DMC to the PmPPC440 (see Fig. 7-8 for DMC location):

1. Remove the protective vinyl caps from the screws.
2. Line up the screws with the threaded holes on the bezel from the bottom side of the PmPPC440.
3. Snap the connectors (P1) together and secure the mounting screws through the standoffs on the DMC to the PmPPC440.

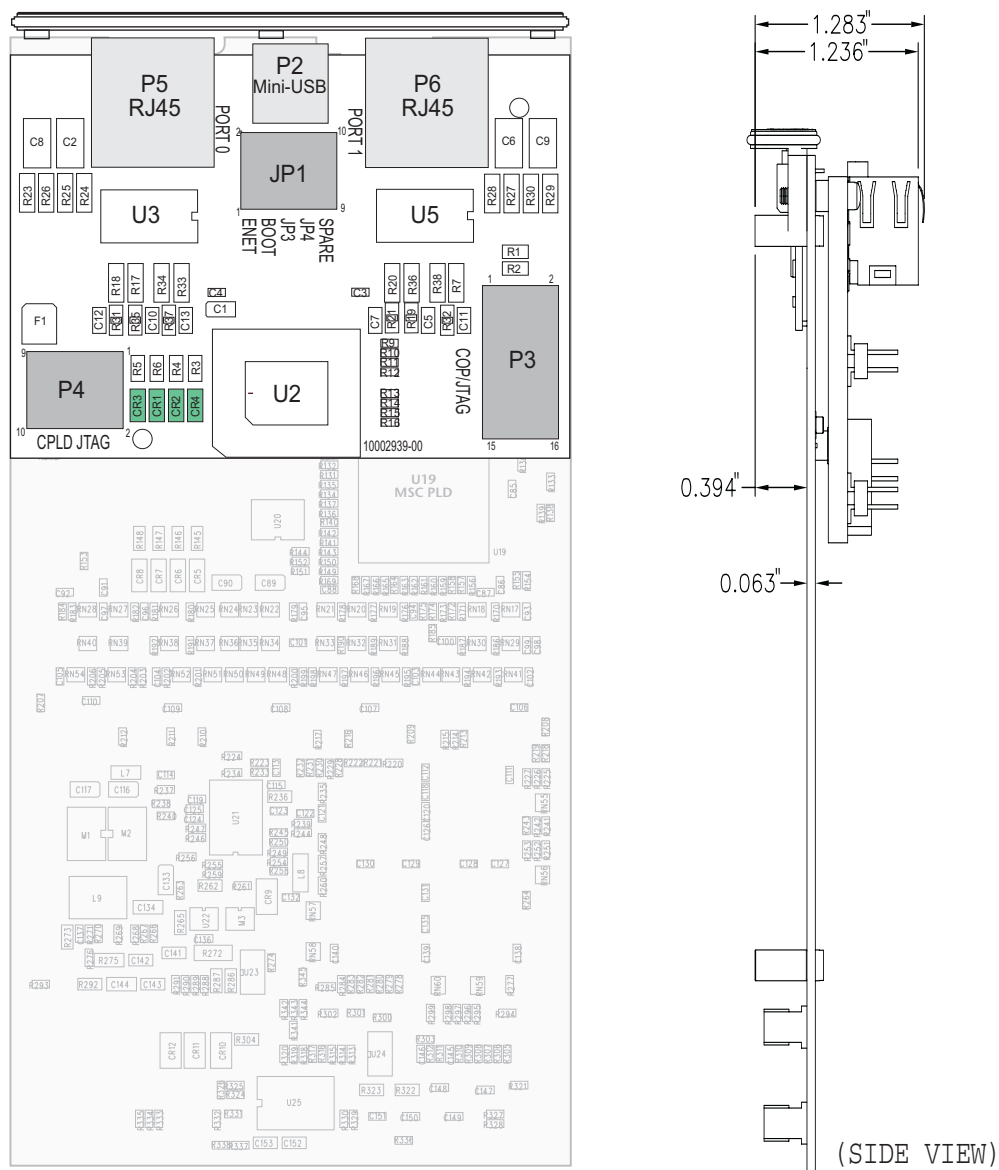


Figure 7-8. DMC Location on PmPPC440

7.6 Troubleshooting

In case of difficulty, use this checklist:

- ☐ Be sure the PmPPC440 module is seated firmly on the PMC and that the PMC host is seated firmly in the card cage.
- ☐ Verify the boot jumper and Ethernet (ENET) settings (see Fig. 7-7).
- ☐ If booting from EEPROM (U2), make sure the device is properly oriented in the socket.
- ☐ Be sure the system is not overheating.
- ☐ Check your power supply for proper DC voltages. If possible, use an oscilloscope to look for excessive power supply ripple or noise (over 50 mV_{pp} below 10 MHz).

7.6.1 Technical Support

After you have checked all of the above items, call our Technical Support department at (800) 327-1251 (or send e-mail to support@artesyndcp.com). Please have the following information handy:

- the DMC serial number
- board assembly/configuration description from the sticker on the DMC board
- the PmPPC440 serial number
- the baseboard model number and monitor revision level (if applicable)
- version and part number of the operating system (if applicable)

7.7 Service Information

If you plan to return the board to Artesyn Communication Products for service, call (800) 356-9602 and ask for our Test Services department (or send e-mail to serviceinfo@artesyndcp.com) to obtain a return merchandise authorization (RMA) number. We will ask you to list which items you are returning and the board serial number, plus your purchase order number and billing information if your DMC is out of warranty. Contact our Test Services department for any warranty questions. If you return the board, be sure to enclose it in an antistatic bag, such as the one in which it was originally shipped. Send it prepaid to:

Artesyn Communication Products
Test Services Department
8310 Excelsior Drive
Madison, WI 53717

RMA # _____

Please put the RMA number on the outside of the package so we can handle your problem efficiently. Our service department cannot accept material received without an RMA number.

8

Monitor

The PmPPC440 monitor is based on the Embedded PowerPC Linux Boot Project (PPCBoot) boot program, available under the GNU General Public License (GPL). For instructions on how to obtain the source code for this GPL program, please visit <http://www.artesyncp.com>, send an e-mail to support@artesyncp.com, or call Artesyn at 1-800-327-1251.

This chapter describes the monitor's basic features, basic operation, and configuration sequences. This chapter also serves as a reference for the monitor commands and functions.

8.1 Monitor Features

The PmPPC440 monitor uses a command-line interface. This section describes these features, as well as the start-up display.

8.1.1 Start-Up Display

At power-up or after a reset, the monitor runs diagnostics and reports the results in the start-up display, see Fig. 8-1.

During the power-up sequence, the monitor configures the board according to the environment variables (see Section 8.13) and the settings in the Board Configuration registers (see Section 2.7). If the configuration indicates that autoboot is enabled, the monitor attempts to load the application from the specified device. If the monitor is not configured for autoboot, or a failure occurs during power-up, the monitor enters normal command-line mode.

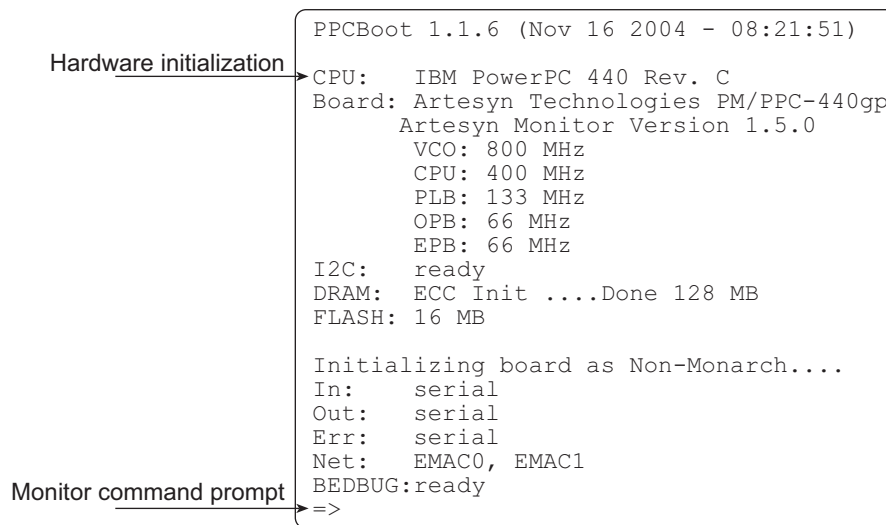


Figure 8-1. Monitor Start-up Display

8.1.2 Auto-repeat

After a command has been entered, it can be re-executed simply by pressing the ENTER or RETURN key.

8.1.3 TFTP booting

Application images can be loaded via Ethernet into the PmPPC440's memory using the TFTP protocol.

8.1.4 Auto-booting

Specific boot commands can be stored in the environment and executed automatically after reset.

8.1.5 Flash Programming

Application images can be written into Flash via the PPCBoot command line.

8.2 Basic Operation

The PmPPC440 monitor performs various configuration tasks upon power-up or reset. This section describes the monitor operation during initialization of the PmPPC440 board.

8.2.1 Power-up/Reset Sequence

At power-up or board reset, the PmPPC440 performs the following:

1. The PPC440GP processor has its “I²C Bootstrap Operation” enabled, which forces it to pre-load several system registers from an external serial ROM device located on the I²C bus. The ROM contains essential power-up, system, and customer information, such as PLL/clock divisors, ROM width and location (local or PCI), Ethernet mode selection, etc.
2. The monitor shall boot from effective address FFFF,FFFC₁₆ and branches backward four kilobytes (1000₁₆) to begin memory management initialization.
3. The monitor configures the PPC440GP memory management unit (MMU), including the cache and TLB entries, which control memory, access, and storage attributes.
4. The monitor initializes on-board devices as follows:

(These are ***board_init_f()*** function operations while PPCBoot is in flash.)

- On-chip SRAM to store preliminary C stack
- Chip selects for each peripheral device on-card
- On-chip timebase
- Board environment variables (read from NVMemory – I²C)
- Serial/console port
- Display the version of PPCBoot, CPU with revision, and board information
- I²C ports –initialize and display result
- SDRAM – initialize and display result
- Move the C stack into RAM
- Relocate PPCBoot from EEPROM to RAM

5. Clear the C BSS section.
6. The monitor initializes on-board devices as follows:

(These are ***board_init_f()*** function operations while PPCBoot is in RAM.)
 - Load all PPCBoot commands into RAM.
 - Copy exception vector table into low RAM.
 - Determine and display soldered flash size.
 - Allocate C 'malloc' memory area.
 - Relocate environment function pointers into RAM.
 - Configure and display PCI information.
 - Re-initialize I²C ports in RAM.
 - Re-initialize serial/console in RAM and display where standard in, out, and error are directed.
 - Configure the system call table.
 - Reset the Ethernet PHY device.
 - Set the 'kgdb' (allows PPCBoot to enter GNU GDB debug mode).
 - Configure and enable interrupts.
 - Configure the PPC440GP internal debug mode registers and display status.
 - Configure Ethernet and display status.
 - Start the command line interpreter.

8.2.2 Monitor SDRAM Usage

PPCBoot locates its stack, uninitialized data, and code at the very end of SDRAM. The exact address varies with the amount of installed memory.

CAUTION. Any writes to this area can cause unpredictable operation of the monitor.

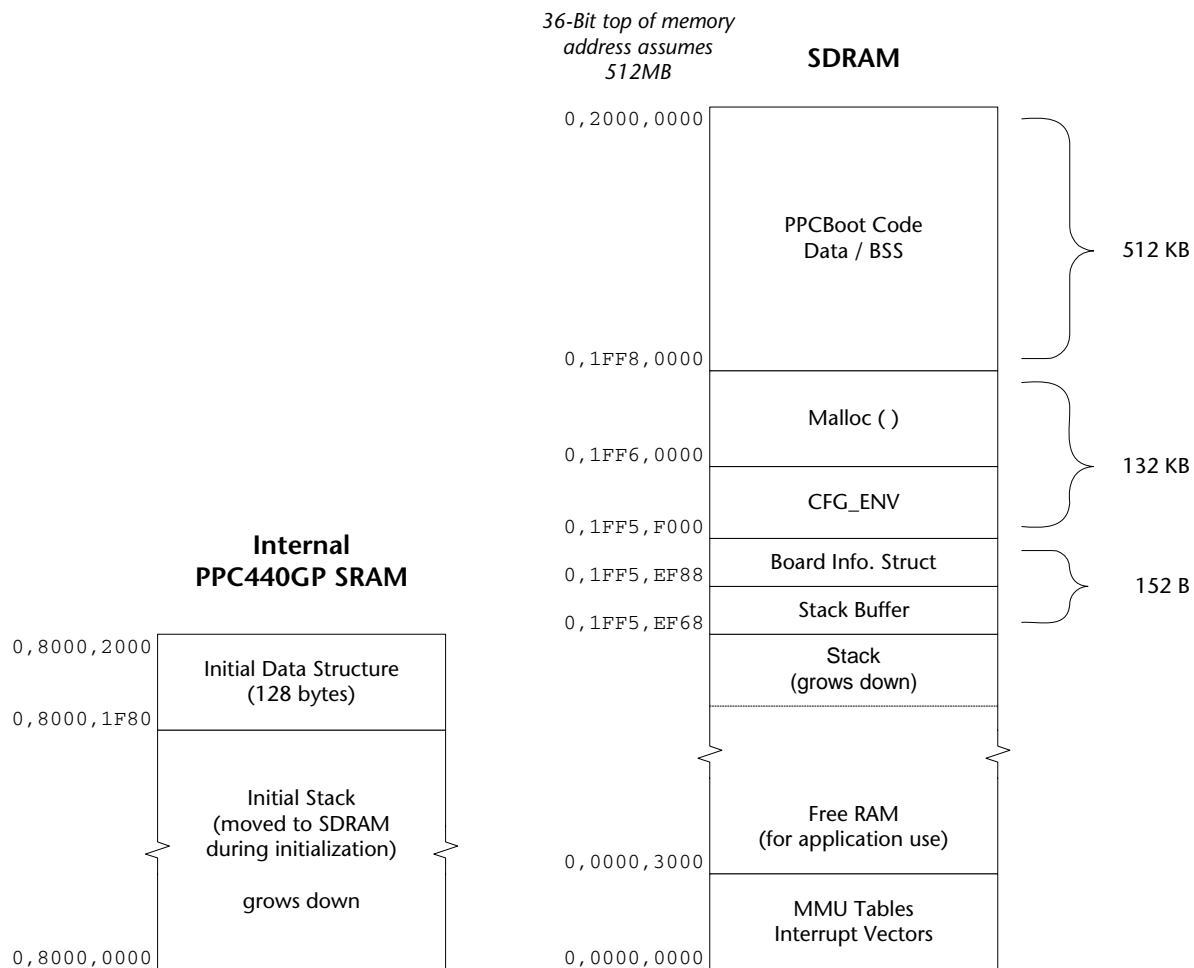


Figure 8-2. Monitor Memory Usage

8.3 Monitor Recovery

This section describes how to recover the monitor, given one or more of the following conditions:

- If there is no console output, the monitor may be corrupted and need recovering.
- If the monitor still functions, but is not operating properly, then you may need to reset the environment variables.
- If you are having Ethernet problems in the monitor, you may need to set the serial number, since the MAC address is calculated from the serial number variable.

8.3.1 Corrupted Monitor in Soldered Flash

This procedure describes how to recover from corrupted monitor code in the soldered flash memory.

1. First, attach a DMC module to the PmPPC440 PMC module. Make sure that a monitor ROM device is installed in the PLCC socket on DMC module and set the boot jumper on the DMC module to boot the PmPPC440 from the socket ROM device (see section 7.3).
2. Issue the following command, where *serial_number* is the board's serial number, at the monitor prompt:
`=> moninit serial_number`

This command re-initializes the environment parameters, re-programs the bootstrap I²C parameters, erases the flash (where the monitor resides), and copies the monitor code from the PLCC socket to the soldered flash.

3. Turn off power to the board.
4. Remove the DMC module and re-apply power.

The monitor should now boot correctly.

8.3.2 Corrupted Bootstrap I²C Parameters

The IBM PPC440GP processor requires initialization parameters loaded from the bootstrap I²C nonvolatile memory device. By default, this device is write-protected. (However, it can be unprotected by writing to a PLD register.) If the bootstrap parameters are not properly initialized or are corrupted, the PmPPC440 board may become unstable. The following procedure describes how to recover from this situation:

1. If the board does not power-up correctly, attach the DMC module with a monitor ROM installed in its PLCC socket. Set the module to use the default bootstrap parameters by ensuring that jumper JP4 is installed (see section 7.3).
2. To re-initialize the bootstrap parameters, type the following from the monitor prompt:

```
=> prog_i2c
```

3. Turn off power to the board.
4. Remove the DMC module and re-apply power.

The monitor should now boot correctly.

8.3.3 Resetting Environment Variables

To reset the monitor's environment variables to their factory default values, execute the following commands, where *serial_number* is the board's serial number:

```
=> initenv serial_number  
=> initenv 1053
```

Upon completion, you can view the updated environment variables by typing the following command:

```
=> printenv
```

Please refer to section 8.10 for a description of the monitor environment variables.

8.4 Monitor Command Reference

This section describes the syntax and typographic conventions for the PmPPC440 monitor commands. Subsequent sections in this chapter describe individual commands, which fall into the following categories: boot, memory, Flash, environment variables, test, and other commands.

8.4.1 Command Syntax

The monitor uses the following basic command syntax:

<Command> <argument 1> <argument 2> <argument 3>

- The command line accepts three different argument formats: string, numeric, and symbolic. All command arguments must be separated by spaces with the exception of argument flags, which are described below.
- Monitor commands that expect numeric arguments assume a hexadecimal base.
- All monitor commands are case sensitive.
- Some commands accept flag arguments. A flag argument is a single character that begins with a period (.). There is no white space between an argument flag and a command. For example, **md.b 80000** is a valid monitor command, while **md .b 80000** is not.
- Some commands may be abbreviated by typing only the first few characters that uniquely identify the command. For example, you can type **ver** instead of **version**. However, commands cannot be abbreviated when accessing on-line help. You must type help and the full command name.

8.4.2 Typographic Conventions

In the following command descriptions, `Courier New` font is used to show the command format. Square brackets `[]` enclose optional arguments, and *Italic* type indicates that you must substitute your own selection for the italicized text.

8.5 Boot Commands

The boot commands provide facilities for booting application programs and operating systems from various devices.

8.5.1 bootm (boot memory)

The **bootm** command boots an application image stored in memory, passing any entered arguments to the called application. When booting a Linux kernel, *arg* can be the address of an initrd image. If *addr* is not specified, the environment variable **loadaddr** is used as the default.

DEFINITION

```
bootm [addr [arg ...]]
```

8.5.2 bootp (boot BootP/TFTP)

The **bootp** command boots an image via a network connection using the BootP/TFTP protocol. If *loadaddress* or *bootfilename* is not specified, the environment variables **loadaddr** and **bootfile** are used as the default.

DEFINITION

```
bootp [loadAddress] [bootfilename]
```

8.5.3 tftpboot (boot TFTP)

The **tftpboot** command boots an image via a network connection using the TFTP protocol. The environment variable's *ipaddr* and *serverip* are used as additional parameters to this command. If *loadaddress* or *bootfilename* is not specified, the environment variables **loadaddr** and **bootfile** are used as the default.

DEFINITION

```
tftpboot [loadAddress] [bootfilename]
```

8.5.4 rarpboot (boot RARP/TFTP)

The **rarpboot** command boots an image via a network connection using the RARP/TFTP protocol. If *loadaddress* or *bootfilename* is not specified, the environment variables **loadaddr** and **bootfile** are used as the default.

DEFINITION

```
rarpboot [loadAddress] [bootfilename]
```

8.5.5 bootd (boot default)

Execute the command stored in the “**bootcmd**” environment variable.

8.5.6 bootelf (boot .elf image)

The **bootelf** command boots an .elf image stored in memory. If *addr* is not specified, the environment variable **loadaddr** is used as the default.

DEFINITION

```
bootelf [addr]
```

8.5.7 bootvx (boot VxWorks)

The **bootm** command boots an application image stored in memory, passing any entered arguments to the called application. When booting a VxWorks kernel, *arg* can be the address of an initrd image. If *addr* is not specified, the environment variable **loadaddr** is used as the default.

DEFINITION

```
bootm [addr [arg ...]]
```

8.6 File Load Commands

8.6.1 loads (load S-record)

The **loads** command loads an S-Record file over the serial port. The command takes two optional parameters:

- | | |
|-----------------|---|
| offset | The address offset parameter allows the file to be stored in a location different than what is indicated within the S-Record file by adding the value [off] to the file's absolute address. |
| baudrate | The baudrate parameter allows the file to be loaded at [baud] instead of the monitor's console baudrate. |

The file is not automatically executed, the **loads** command only loads the file into memory.

DEFINITION

```
loads [off] [baud]
```

8.6.2 loadb (load binary)

The **loadb** command loads a binary file over the serial port. The command takes two optional parameters:

- | | |
|-----------------|---|
| offset | The address offset parameter allows the file to be stored in a location different than what is indicated within the binary file by adding the value [off] to the file's absolute address. |
| baudrate | The baudrate parameter allows the file to be loaded at [baud] instead of the monitor's console baudrate. |

The file is not automatically executed, the **loadb** command only loads the file into memory.

DEFINITION

```
loadb [off] [baud]
```

8.7 Memory Commands

The memory commands provide facilities for manipulating specific regions of memory. For some memory commands, the data size is determined by the following flags:

- .b** This is for data in 8-bit bytes.
- .w** This is for data in 16-bit words.
- .l** This is for data in 32-bit long words.

These flags are optional arguments and describe the objects on which the command operates. The memory commands default to 32-bit long words if no flag is specified. Numeric arguments are in hexadecimal.

8.7.1 md (memory display)

The command **md** displays the contents of memory starting at *address*. The number of objects displayed can be defined by an optional third argument, *# of objects*. The memory's numerical value and its ASCII equivalent is displayed.

DEFINITION

```
md [.b, .w, .l] address [# of objects]
```

EXAMPLE

In this example, the **md** command is used to display thirty-two 16-bit words starting at the physical address 0x80000.

```
=> md.w 80000 20
00080000: ffff ffff ffff ffff ffff ffff ffff ffff .....
00080010: ffff ffff ffff ffff ffff ffff ffff ffff .....
00080020: ffff ffff ffff ffff ffff ffff ffff ffff .....
00080030: ffff ffff ffff ffff ffff ffff ffff ffff .....
```

8.7.2 mw (memory write)

The command **mw** writes *value* to memory starting at *address*. The number of objects modified can be defined by an optional fourth argument, *count*.

DEFINITION

```
mw [.b, .w, .l] address value [count]
```

EXAMPLE

In this example, the **mw** command is used to write the value 0xabba three times starting at the physical address 0x80000.

```
=> mw.w 80000 abba 3
=> md 80000
00080000: abbaabba abbaffff ffffffff ffffffff .....
00080010: ffffffff ffffffff ffffffff ffffffff .....
00080020: ffffffff ffffffff ffffffff ffffffff .....
00080030: ffffffff ffffffff ffffffff ffffffff .....
00080040: ffffffff ffffffff ffffffff ffffffff .....
00080050: ffffffff ffffffff ffffffff ffffffff .....
00080060: ffffffff ffffffff ffffffff ffffffff .....
00080070: ffffffff ffffffff ffffffff ffffffff .....
```

8.7.3 mm (memory modify – incrementing address)

The **mm** command modifies memory one object at a time. Once started, the command line prompts for a new value at the starting address. After a new value is entered, pressing ENTER auto-increments the address to the next location. Pressing ENTER without entering a new value leaves the original value for that address unchanged. To exit the **mm** command, enter a non-valid hexadecimal value (such as x) followed by ENTER.

DEFINITION

```
mm [.b, .w, .l] address
```

EXAMPLE

In this example, the **mm** command is used to write random 8-bit data starting at the physical address 0x80000.

```
=> mm.b 80000
00080000: ff ? 12
00080001: ff ? 23
00080002: ff ? 34
00080003: ff ? 45
00080004: ff ?
00080005: ff ? x
=> md.b 80000 6
00080000: 12 23 34 45 ff ff .#4E
=>
```

8.7.4 nm (memory modify – constant address)

The **nm** command modifies a single object repeatedly. Once started, the command line prompts for a new value at the selected address. After a new value is entered, pressing ENTER modifies the value in memory and then the new value is displayed. The command line then prompts for a new value to be written at the

same address. Pressing ENTER without entering a new value leaves the original value unchanged. To exit the **nm** command, enter a non-valid hexadecimal value (such as x) followed by ENTER.

DEFINITION

```
nm [.b, .w, .l] address
```

8.7.5 cp (copy memory)

The **cp** command copies *count* objects located at the *source* address to the *target* address. It is also used to copy memory to the soldered flash.

NOTE. If the *target* address is located in the range of the Flash device, it will program the Flash with count objects from the source address. The **cp** command does not erase the Flash region prior to copying the data. The Flash region must be manually erased using the erase command prior to using the **cp** command.

DEFINITION

```
cp [.b, .w, .l] source target count
```

EXAMPLE

In this example, the **cp** command is used to copy 0x1000, 32-bit values from address 0x100000 to address 0x80000.

```
=> cp 100000 80000 1000
```

8.7.6 cmp (compare memory)

The **cmp** command compares *count* objects between *addr1* and *addr2*. Any differences are displayed on the console display.

DEFINITION

```
cmp [.b, .w, .l] addr1 addr2 count
```

8.8 I²C Device Commands

8.8.1 imd (I²C memory display)

The command **imd** displays the contents of memory for an I²C device starting at *address*. The number of objects displayed can be defined by an optional third argument, *# of objects*. The memory's numerical value and its ASCII equivalent is displayed.

DEFINITION

```
imd chip address [.0, .1, .2] [# of objects]
```

8.8.2 imm (I²C memory modify – incrementing address)

The **imm** command modifies memory for an I²C device one object at a time. Once started, the command line prompts for a new value at the starting address. After a new value is entered, pressing ENTER auto-increments the address to the next location. Pressing ENTER without entering a new value leaves the original value for that address unchanged. To exit the **imm** command, enter a non-valid hexadecimal value (such as x) followed by ENTER.

DEFINITION

```
imm chip address [.0, .1, .2]
```

8.8.3 inm (I²C memory modify – constant address)

The **inm** command modifies a single object repeatedly for an I²C device. Once started, the command line prompts for a new value at the selected address. After a new value is entered, pressing ENTER modifies the value in memory and then the new value is displayed. The command line then prompts for a new value to be written at the same address. Pressing ENTER without entering a new value leaves the original value unchanged. To exit the **inm** command, enter a non-valid hexadecimal value (such as x) followed by ENTER.

DEFINITION

```
inm chip address [.0, .1, .2]
```

8.8.4 imw (I²C memory write)

The command **imw** writes *value* to memory for an I²C device starting at *address*. The number of objects modified can be defined by an optional fourth argument, *count*.

DEFINITION

```
imw chip address [.0, .1, .2] value [count]
```

8.8.5 icrc32

The **icrc32** command computes a CRC32 checksum for an I²C device on *count* bytes starting at *address*.

DEFINITION

```
icrc32 chip address count
```

8.8.6 iprobe

The **iprobe** command discovers valid I²C devices. It will scan the I²C bus for any valid devices and return the addresses.

DEFINITION

```
iprobe
```

EXAMPLE

```
=> iprobe  
Valid chip addresses: A0 A2 A8 D0  
=>
```

8.8.7 iloop

The **iloop** command executes an infinite loop on an address range.

DEFINITION

```
iloop chip address [.b, .w, .l] number_of_objects
```

8.8.8 isdram

The **isdram** command prints the SDRAM configuration for an SPD device.

DEFINITION

```
isdram chip
```

EXAMPLE

```
=> isdram a2
SPD data revision          0.0
Bytes used                 0x80
Serial memory size         0x100
Memory type                unknown
Row address bits           13
Column address bits        9
Module rows                1
Module data width          72 bits
Interface signal levels    SSTL 2.5
SDRAM cycle time           7.5 nS
SDRAM access time          7.5 nS
EDC configuration          ECC
Self refresh, rate         7.8uS
SDRAM width (primary)      16
EDC width                  16
Min clock delay, back-to-back random column addresses 1
Burst length(s)            8 4 2
Number of banks            4
CAS latency(s)             4 3
CS latency(s)              0
WE latency(s)              1
Module attributes:
Differential clock input
Device attributes:
Upper Vcc tolerance 10%
Lower Vcc tolerance 10%
SDRAM cycle time (2nd highest CAS latency)    10.0 nS
SDRAM access from clock (2nd highest CAS latency) 7.5 nS
SDRAM cycle time (3rd highest CAS latency)     0.0 nS
SDRAM access from clock (3rd highest CAS latency) 0.0 nS
Minimum row precharge                80 nS
Row active to row active min         60 nS
RAS to CAS delay min                 80 nS
Minimum RAS pulse width               45 nS
Density of each row                   128MByte
Command and Address setup             -1.0 nS
Command and Address hold              -1.0 nS
Data signal input setup               +5.0 nS
```

```

Data signal input hold      +5.0 nS
Manufacturer's JEDEC ID    7F A8 00 00 00 00 00 00
Manufacturing Location      01
Manufacturer's Part Number  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00
Revision Code               00 00
Manufacturing Date          02 26
Assembly Serial Number      0B 00 CF 03
Speed rating                PC0
=>

```

8.9 Flash Commands

The Flash commands affect the StrataFlash devices on the PmPPC440 circuit board. There is a maximum of two Flash banks on the PmPPC440 board. The following Flash commands access the individual Flash banks as Flash bank 1 or Flash bank 2. To access the individual sectors within each Flash bank, the sector numbers start at 0 and end at one less than the total number of sectors in the bank. For a Flash bank with 128 sectors, the following Flash commands access the individual sectors as 0 through 127.

8.9.1 flinfo (Flash info)

The **flinfo** command prints out the Flash device's manufacturer, part number, size, number of sectors, and starting address of each sector.

DEFINITION

Print information for all Flash memory banks.

```
flinfo
```

Print information for the Flash memory in bank # *N*.

```
flinfo N
```

8.9.2 erase (Flash erase)

The **erase** command erases the specified area of Flash memory.

DEFINITION

Erase all of the sectors in the address range from *start* to *end*.

```
erase start end
```

Erase all of the sectors *SF* (first sector) to *SL* (last sector) in Flash bank # *N*.

```
erase N:SF[-SL]
```

Erase all of the sectors in Flash bank # *N*.

```
erase bank N
```

Erase all of the sectors in all of the Flash banks.

```
erase all
```

8.9.3 protect (Flash sector protect)

The **protect** command enables or disables the Flash sector protection for the specified Flash sector. For ECI boards, protection is implemented using software only. The protection mechanism inside the physical Flash part is not being used.

DEFINITION

Protect all of the Flash sectors in the address range from *start* to *end*.

```
protect on start end
```

Protect all of the sectors *SF* (first sector) to *SL* (last sector) in FLASH bank # *N*.

```
protect on N:SF[-SL]
```

Protect all of the sectors in Flash bank # *N*.

```
protect on bank N
```

Protect all of the sectors in all of the Flash banks

```
protect on all
```

Remove protection on all of the Flash sectors in the address range from *start* to *end*.

```
protect off start end
```

Remove protection on all of the sectors *SF* (first sector) to *SL* (last sector) in FLASH bank # *N*.

```
protect off N:SF[-SL]
```

Remove protection on all of the sectors in Flash bank # *N*.

```
protect off bank N
```

Remove protection on all of the sectors in all of the Flash banks.

```
protect off all
```

8.10 Environment Parameter Commands

The monitor uses on-board, non-volatile memory for the storage of environment parameters. Environment parameters are stored as ASCII strings with the following format.

<Parameter Name>=<Parameter Value>

Some environment variables are used for board configuration and identification by the monitor. The environment parameter commands deal with the reading and writing of these parameters. Refer to Section 8.13 for a list of monitor environment variables.

8.10.1 printenv (print environment)

The **printenv** command displays all of the environment variables and their current values to the display.

DEFINITION

Print the values of all environment variables.

```
printenv
```

Print the values of all environment variable (exact match) 'name'.

```
printenv name ...
```

8.10.2 setenv (set environment)

The **setenv** command adds new environment variables, sets the values of existing environment variables, and deletes unwanted environment variables.

DEFINITION

Set the environment variable *name* to *value* or adds the new variable *name* and *value* to the environment.

```
setenv name value
```

Removes the environment variable *name* from the environment.

```
setenv name
```

8.10.3 saveenv (save environment)

The **saveenv** command writes the environment variables to non-volatile memory.

DEFINITION

```
saveenv
```

8.10.4 initenv (initialize environment)

The **initenv** command initializes all PmPPC440-specific environment variables to their factory defaults. This command over-writes any existing configurations. The four-digit serial number must be entered with this command.

DEFINITION

```
initenv serial number
```

8.11 Test Commands

8.11.1 um (destructive memory test)

The **um** command is a destructive memory test.

DEFINITION

```
um [.b, .w, .l] base_addr [top_addr]
```

8.11.2 mtest (memory test)

The **mtest** command performs a simple SDRAM read/write test.

DEFINITION

```
mtest [start [end [pattern]]]
```

8.12 Other Commands

8.12.1 go

The **go** command runs an application at address *addr*, passing the optional arguments *arg* to the called application. If *addr* is not specified, the environment variable **loadaddr** is used as the default.

DEFINITION

```
go addr [arg...]
```

8.12.2 run

The **run** command runs the commands in an environment variable *var*.

DEFINITION

```
run var [...]
```

8.12.3 crc32

The **crc32** command computes a CRC32 checksum on *count* bytes starting at *address*.

DEFINITION

```
crc32 address count
```

8.12.4 base

The **base** command prints or sets the address offset for memory commands.

DEFINITION

Displays the address offset for the memory commands.

```
base
```

Sets the address offset for the memory commands to *off*.

```
base off
```

8.12.5 binfo

The **binfo** command displays the Board Information Structure.

DEFINITION

```
binfo
```

8.12.6 iminfo

The **iminfo** command displays the header information for an application image that is loaded into memory as address *addr*. Verification of the image contents (magic number, header, and payload checksums) are also performed.

DEFINITION

```
iminfo addr [addr ...]
```

8.12.7 coninfo

The **coninfo** command displays the information for all available console devices.

DEFINITION

`coninfo`

8.12.8 loop

The **loop** command executes an infinite loop on address range.

DEFINITION

`loop [.b, .w, .l] address number_of_objects`

8.12.9 reset

The **reset** command performs a hard reset of the CPU by writing to the reset register on the board.

DEFINITION

`reset`

8.12.10 echo

The **echo** command echoes *args* to console.

DEFINITION

`echo [args..]`

8.12.11 version

The **version** command displays the monitor's current version number.

DEFINITION

`version`

8.12.12 sleep

The **sleep** command executes a delay of *N* seconds.

DEFINITION

Delay execution for *N* seconds (*N* is a decimal value).

`sleep N`

8.12.13 help

The **help** (or **?**) command displays the online help. Without arguments, all commands are displayed with a short usage message for each. To obtain more detailed information for a specific command, enter the desired command as an argument.

DEFINITION

```
help [command ...]
```

8.13 Environment Variables

Table 8-1. Standard Environment Variables

Variable	Default Value	Description
baudrate	9600	Console baud rate. Valid rates: 300, 600, 1200, 2400, 9600, 19200, 38400
bootcmd	tftpboot; go	Command to be executed after boot
bootdelay	-1	Countdown to bootcmd execute (-1 to disable autoboot)
bootfile	path/file.bin	Path to boot file on server (used with TFTP)
pci_enum	on	PCI enumeration if module is PPMC Monarch (on = enumerate if Monarch, off = never enumerate)
eth1addr	00:80:F9:69:xx:xx	Second Ethernet interface MAC address (xx:xx determined by board serial number)
ethaddr	00:80:F9:69:xx:xx	First Ethernet interface MAC address (xx:xx determined by board serial number)
gatewayIP	0.0.0.0	Gateway IP address
ipaddr	0.0.0.0	Board IP address
loadaddr	80000	Address to which a boot image is loaded
loads_echo	0	Echo characters received during serial download
netmask	0.0.0.0	Board subnet mask
pci_listing	short	Output format for displaying PCI listing (short = brief, long = full)
serverip	0.0.0.0	Boot server IP address
stderr	serial	stderr interface
stdin	serial	stdin interface
stdout	serial	stdout interface

Table 8-2. Optional Environment Variable ^b

Variable	Default Value	Description
ecc	on	Enable/disable ECC protection (on = enable, off = disable)
^b This is not initialized by initenv. (Note: This variable is currently not implemented.)		

8.14 Troubleshooting

To bypass the full board initialization sequence, attach a terminal to the console located on the front of the module. Configure the terminal parameters to be:

9600 bps, no parity, 8 data bits, 1 stop bit

Reset the module while holding down the 's' key. Pressing the 's' key forces a default console configuration and bypasses enabling of caches, ECC, and PCI.

8.15 Download Formats

8.15.1 Binary

The binary download format consists of two parts:

- Magic number (which is 0x12345670) + number of sections
- Information for each section including: the load address (unsigned long), the section size (unsigned long), and a checksum (unsigned long) that is the long-word sum of the memory bytes of the data section

8.15.2 Motorola S-Record

S-Record download uses the standard Motorola S-Record format. This includes load address, section size, and checksum all embedded in an ASCII file.

Index

B

- binary download format 8-25
- block diagram
 - general system 1-3
 - PPC440GP 3-2
 - RTC 4-1
- boot
 - commands, monitor 8-9
 - device JP2 7-10

C

- caution statements
 - writes to monitor area 8-5
- component map
 - DMC 7-2
 - PmPPC440, bottom 2-4
 - PmPPC440, top 2-3
- connectors
 - overview, DMC 7-3
 - overview, module 2-5
 - PMC 5-5
- customer support
 - contacting 1-6
 - service 2-12

D

- development mezzanine card (DMC)
 - circuit board 7-1
 - connectors 7-3–7-9
 - installation 7-12
 - jumpers 7-10
 - LEDs 7-11
 - setup 7-11
 - troubleshooting 7-13

E

- environment parameter commands, monitor 8-20
- environment variables 8-7
- equipment for setup 2-6
- error, parity 5-3
- ESD prevention 2-1
- Ethernet address 6-2
- Ethernet connectors 6-3

F

- file load commands, monitor 8-11
- Flash commands, monitor 8-18

G

- grounding 2-1

I

- installation 2-7, 2-8
 - DMC 7-12
- interrupts, PMC/PCI 5-3

J

- JTAG/COP interface 7-7
- jumpers, DMC 7-10

L

- LEDs
 - control register 2-10
 - DMC 7-11
 - module front panel 2-2

M

memory

- commands, monitor 8-12
- flash 3-4
- serial EEPROM 3-5
- SO-DIMM 3-5

monitor

- auto-booting 8-2
- auto-repeat 8-2
- binary download format 8-25
- boot commands 8-9
- command syntax 8-8
- environment parameter commands 8-20
- environment variables 8-24
- file load commands 8-11
- Flash commands 8-18
- Flash programming 8-2
- memory commands 8-12
- Motorola S-record 8-25
- other commands 8-21
- PPCBOOT 8-1
- recovery 8-6
- start-up display 8-2
- test commands 8-21
- TFTP booting 8-2
- troubleshooting 8-25
- typographic conventions 8-9
- version number 2-5

monitor commands

- base 8-22
- bdinfo 8-22
- bootd (boot default) 8-10
- bootelf (boot .elf image) 8-10
- bootm (boot memory) 8-9
- bootp (bootBootP/TFTP) 8-9
- bootvx (boot VxWorks) 8-10
- cmp (compare memory) 8-14
- coninfo 8-23
- cp (copy memory) 8-14
- crc32 8-22
- echo 8-23
- erase (Flash erase) 8-18
- flinfo (Flash info) 8-18
- go 8-21
- help (or '?') 8-24
- I2C crc32 8-16
- I2C DRAM config. 8-17
- I2C loop 8-17
- I2C probe 8-16
- imd (I2C memory display) 8-15
- iminfo 8-22
- imm (I2C memory modify-incrementing address) 8-15

- imw (I2C memory write) 8-15
 - initenv (initialize environment) 8-21
 - loadb (load binary) 8-11
 - loads (load S-record) 8-11
 - loop 8-23
 - md (memory display) 8-12
 - mm (memory modify-incrementing address) 8-13
 - mtest (memory test) 8-21
 - mw (memory write) 8-12
 - nm (I2C memory modify-constant address) 8-15
 - nm (memory modify-constant address) 8-13
 - printenv (print environment) 8-20
 - protect (Flash sector protect) 8-19
 - rarpboot (boot RARP/TFTP) 8-10
 - reset 8-23
 - run 8-22
 - saveenv (save environment) 8-20
 - setenv (set environment) 8-20
 - sleep 8-23
 - tftpboot (boot TFTP) 8-9
 - um (destructive memory test) 8-21
 - version 8-23
- monitor recovery 8-6

N

- notation conventions 1-5

P

PMC/PCI

- interrupts 5-3
- signal descriptions 5-2

R

- real-time clock
 - block diagram 4-1
 - operation 4-2
- references, manuals, and data books 1-5
- regulatory certifications 1-4
- reset 2-9
 - PCI signal 5-4
- returning boards 2-13, 7-13
- RMA number 2-13, 7-14

S

- serial I/O 3-6
- serial number

- DMC 7-2
 - location 2-5
 - user ROM 2-5, 7-3
- service information 2-13, 7-13
- setup
 - DMC requirements 7-11
 - module requirements 2-6
- signal descriptions
 - DMC 7-5
 - PCI 5-2
 - PCI expansion 5-4
- signals
 - 66MHz enable, PCI 5-3
 - ACK64*, PCI 5-2
 - ADDRESS and DATA, PCI 5-2
 - bus command, PCI 5-2
 - byte enable, PCI 5-2
 - clock, PCI 5-2
 - cycle frame, PCI 5-2
 - device select, PCI 5-2
 - grant, PCI 5-3
 - initialization device select, PCI 5-3
 - initiator ready, PCI 5-3
 - lock, PCI 5-3
 - Monarch, PCI 5-3
 - parity error, PCI 5-3
 - parity, PCI 5-3
 - PCIXCAP, PCI 5-3

- PMC interrupt, PCI 5-3
- power management, PCI 5-3
- present, PCI 5-3
- ready, PCI 5-2
- REQ64*, PCI 5-2
- REQ64, PCI 5-3
- request, PCI 5-4
- reset output, PCI 5-3
- reset, PCI 5-4
- stop, PCI 5-4
- systems error, PCI 5-4
- target ready, PCI 5-4
- specifications
 - environmental 2-6
 - mechanical 2-2, 7-1
 - power 2-6
- static control 2-1

T

- technical references 1-5
- technical support 2-12, 7-13
- terminology 1-5
- test commands, monitor 8-21
- troubleshooting 2-12
 - DMC 7-13
 - monitor 8-25